Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет»

Е.В. Мещерина

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Учебно-методическое пособие

Рекомендовано ученым советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по образовательным программам высшего образования по направлениям подготовки 02.03.01 Математика и компьютерные науки, 02.03.02 Фундаментальная информатика и информационные технологии, специальности 10.05.01 Компьютерная безопасность

Оренбург
2019

УДК 004.8(075.8)
ББК 32.813я73
М 56

Рецензент – доцент, кандидат физико-математических наук В.В. Носов

Мещерина, Е.В.
М 56 Системы искусственного интеллекта [Электронный ресурс]: учебно-методическое пособие / Е.В. Мещерина; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2019. – 96 с.
ISBN 978-5-7410-2315-0

Учебно-методическое пособие содержит теоретический материал и задания для лабораторных работ. В нем освещены основы программирования на одном из самых распространенных логических языков – Prolog, основы программирования нейронных сетей, знакомство с нечеткими множествами и нечеткой логикой. Каждая лабораторная работа включает в себя задания для лабораторных работ и вопросы для самоконтроля.

Учебно-методическое пособие предназначено для обучающихся по образовательным программам высшего образования по направлениям подготовки 02.03.01 Математика и компьютерные науки, 02.03.02 Фундаментальная информатика и информационные технологии, специальности 10.05.01 Компьютерная безопасность при изучении дисциплин «Системы искусственного интеллекта», «Интеллектуальные системы», «Основы искусственного интеллекта» соответственно.

УДК 004.8(075.8)
ББК 32.813я73

ISBN 978-5-7410-2315-0

© Мещерина Е.В., 2019
© ОГУ, 2019

Содержание

Введение

Данное учебно-методическое пособие содержит теоретический материал и задания для лабораторных работ. В нем освещены основы программирования на одном из самых распространенных логических языков –Prolog, основы программирования нейронных сетей, знакомство с нечеткими множествами и нечеткой логикой. Каждая лабораторная работа включает в себя задания для лабораторных работ, а также вопросы для самоконтроля.

Язык Prolog и логическое программирование широко используются для создания баз знаний, экспертных систем и оболочек экспертных систем, исследований в сфере искусственного интеллекта на основе логических моделей баз знаний и логических процедур вывода и принятия решений. Нейронные сети используются в сферах обработки графики, математического моделирования, статистического анализа, прогнозирования и т.п. Нечеткие множества и нечеткаялогикаширокоприменяются при построении экспертных систем, систем поддержки принятия решений, систем управления производством.

Каждая тема содержит краткие теоретические сведения с примерами реализации программ, задания для выполнения лабораторных работ, список теоретических вопросов для собеседования при защите работы.

Учебно-методическое пособие предназначено для обучающихся по программам высшего образования по направлениям подготовки 02.03.01 Математика и компьютерные науки, 02.03.02 Фундаментальная информатика и информационные технологии, специальности 10.05.01 Компьютерная безопасность при изучении дисциплин «Системы искусственного интеллекта», «Интеллектуальные системы», «Основы искусственного интеллекта» соответственно.

1 Введение в средупрограммирования Prolog

Пролог (“PROgramminginLOGic”) – это язык логического программирования, предназначенный для работы с символьной информацией, основывающийся на логике предикатов. Он широко используется для решения задач, в которых фигурируют объекты и отношения между ними.

Он используется в таких областях, как система обработки естественных языков, экспертные системы, искусственный интеллект и др.

Рассмотрим некоторые особенности языка Пролог.

– Пролог оперирует с предикатами. Одноместные предикаты определяют свойства их аргументов, многоместные предикаты определяют отношения между аргументами.

– Программа, написанная на Прологе, – это набор утверждений. Каждое утверждение заканчивается точкой. Утверждения еще называют предложениями.

– Утверждения на Прологе бывают трех типов: факты, правила и вопросы. Факты – этоутверждения, которыеявляютсявсегдаверными.

Пример 1. мебель(стол), животное(собака).

Правила – это утверждения, истинность которых зависит от некоторых условий.

Пример 2.животное(X):-имеет_шерсть(X).

С помощью вопросов мы можем спросить систему о том, какие утверждения являются истинными. Обычно вопрос записывается так: за знаком ?- следует ряд хвостовых целевых утверждений.

– Предложения Пролога состоят из головы и тела. Тело – это список целей, разделенных запятыми или точкой запятой.Последовательностьцелей, перечисляемаячерез:

- ,означает конъюнкцию целей (логическое И);
- ; означает дизъюнкцию целей (логическое ИЛИ).

Тогда, факты – это утверждения, имеющие пустое тело, вопросы имеют только тело, правила имеют голову и (непустое) тело. В правилахголова и телоразделенысимволами:-. (см. Пример 2)

– Prolog может оперировать с аргументами следующих типов:

1) атом – в его роли может быть имя, название предмета, он может быть заключен в одинарные кавычки или просто записан с малой буквы; также атомы могут представлять собой цепочки следующих символов:

- прописные буквы русского или латинского алфавита A, B, ..., Z, А, Б, В,…;
- строчные буквы русского и латинского алфавита a, b, ..., z, а, б, в,…;
- цифры 0, 1, 2, ..., 9;
- специальные символы, к примеру, такие как+ - * / = : . & _ ~.

Атомы можно составлять тремя способами:

- из цепочки букв, цифр и символа подчеркивания _, начиная такую цепочку со строчной буквы:

○ nPHn

○ x37

○ a_

- изспециальныхсимволов:

○ <--->

○ ======>

○ ...

○ .

○ ...

○ : : =

При таком составлении атомов необходимо соблюдать осторожность, так как некоторые из цепочек специальных символов в Прологе имеют определенный смысл. Например, символы :- .

- из цепочки символов, заключенных в одинарные кавычки. Это удобно, когда в роли атома выступает имя или название, начинающееся с прописной буквы, чтобы Пролог-программа не воспринимала его в качестве переменной:
    - 'АНСа'

2) списки (записываются в [], например: оценки([5,4,5,5,3,5]);

3) переменная – это цепочка символов, может быть записана с прописной буквы, например: X,Человек; с символа подчеркивания, например: _X.

Если переменная используется в утверждении только один раз, то не обязательно задавать ей имя. Вместо имени можно использовать один символ подчеркивания. Например: умный(_):-много_читает(_).

4) составной терм – это терм, состоящий из атома-функтора, за которым идет несколько аргументов, (например: мебель(стол), родитель(X,У)).

– Пролог-система рассматривает вопросы как цели, к достижению которых нужно стремиться. Ответ на вопрос может оказаться или положительным или отрицательным в зависимости от того, может ли быть соответствующая цель достигнута или нет. Если на вопрос существует несколько ответов, пролог-система найдет столько из них, сколько пожелает пользователь.

– По ходу вычислений переменной может быть присвоено имя или значение. Это значит, что переменная конкретизирована. Еще этот процесс называют унификацией.

Пример 3.То, что Петр является родителем Димы, можно записать на Прологе так:

| родитель(петр,дима).

или

| родитель (‘Петр’, ‘Дима’).

Можно построить дерево родственных отношений. В этом случае программа будет выглядеть так:

```
родитель(петр, дима).
родитель(петр, оля).
родитель(анна, дима).
родитель(анна, оля).
родитель(оля, сергей).
родитель(дима, макс).
```

После составления кода, программе можно задать следующие вопросы:

1. Является ли Петр родителем Оли?

```
| ?- родитель(петр, оля).
```

Так как этот факт записан в программе, система должна ответить:

```
| yes.        (да)
```

или

```
| true.       (истина)
```

2. Является ли Оля родителем Димы?

```
| ?- родитель(оля, дима).
```

Тогда мы получим ответ:

```
| no.         (нет)
```

или

```
| false.      (ложь)
```

так как в программе нет правила, утверждающего этот факт.

3. Является ли Алекс родителем Макса?

```
| ?- родитель(алекс, макс).
```

Ответ программы будет также отрицательный, потому что имя Алекс в программе не упоминается.

4. Кто является родителем Макса?

```
| ?- родитель(X, макс).
```

или

```
| ?- родитель(Кто, макс).
```

Ответ будет таким:

```
| X = дима.
```

или

| Кто =дима.

5. Можно задавать вопросы, на которые у системы имеется несколько вариантов ответа. Например, «Кто дети Петра?»:

| ?- родитель(петр,Кто).

Сначала система выдаст первое решение:

| Кто =дима.

Если мы хотим получить другие решения, то нужно сообщить это системе (во многих реализациях Prolog для этого нужно нажать ;):

| Кто = оля.

Если мы продолжим запрос других решений, то получим ответ “false”, поскольку других решений у системы нет.

6. Мы можем формулировать и более общие вопросы: “Кто чей родитель?”. Сначала переформулируем этот вопрос на язык предикатов: Найти такие X и У, что X является родителем У.

На Прологе это можно записать так:

| ?- родитель(X, Y).

Программа будет находить по очереди все пары вида "родитель-ребенок". По мере того, как мы будем запрашивать у системы новые решения, они будут выводиться на экран друг за другом до тех пор, пока все не будут найдены. Ответы выводятся следующим образом:

X =петр

Y =дима;

X =петр

Y =оля;

X = анна

Y =дима;

...

Если вместо точки с запятой нажать точку, то поток решений будет остановлен.

Пример 4. Известно, что дедушка человека – это папа его мамы или папа его папы.

Соответствующие правила будут иметь вид:

дедушка(X,Y):–папа(X,Z),мама(Z,Y).

дедушка(X,Y):–папа(X,Z),папа(Z,Y).

Первое правило сообщает, что X является дедушкойY, если существует такой Z, что X является папойZ, а Z т – мамой Y. Второе правило сообщает, что X является дедушкойY, если существует такой Z, что X является папой Z, а Z– папой Y.

В данном примере X,Y и Z – это переменные.

Рассмотрим задачу об “интересной жизни”. [3]

Все небедные и умные люди счастливы. Человек, читающий книги, - неглуп. Джонумеет читать и является состоятельным человеком. Счастливые люди живут интересной жизнью. Можно ли указать человека, живущего интересной жизнью?

На языке Prolog эту задачу можно задать так:

```
happy(X):-rich(X), clever(X).
clever(X):-read(X).
rich(tom).
read(tom).
interesting_life(X):-happy(X).
?-interesting_life(X).
```

Решением задачи будет ответ: человек, живущий интересной жизнью – это Том. Система сформулирует ответ так: X=tom.

1.1 Лабораторная работа № 1. Формализация задачи в среде Prolog

Цель работы:Научиться формализовывать задачи в среде Prolog.

Задание:Реализовать задачу с использованием среды Prolog.

1) Запишите на языке Prolog следующие предложения:

а)Дети любят гулять, если на улице тепло.

б) Всякий, кто имеет ребенка, – счастлив.

в) Все умные люди любят читать.

г) Если на улице тепло, то Оля идет в парк.

д) Известно, что любой удачливый или старательный студент может сдать все экзамены.

2) Коля любит футбол. Маша любит фигурное катание. Петя любит хоккей. Катя любит плавание. Саша любит то, что любит Маша. Что любит Саша? Кто любит фигурное катание?

3) Любой студент, который сдает экзамен по математике и выигрывает в лотерею, – счастлив. Известно, что любой удачливый или старательный студент может сдать все экзамены. Джон не относится к числу старательных студентов, но достаточно удачлив. Любойудачливыйстудентвыигрывает в лотерею. СчастливлиДжон?

Вопросы для собеседования при защите работы:

1. Классификацияпредставлениязадач.
2. Логическиемодели.
3. Сетевыемодели.
4. Продукционныемодели.
5. Сценарии.

2 Построение простой вопросно-ответной схемы в Prolog. Работа со списками

В Прологе вопрос записывается почти так же, как и факт, за исключением того, что перед ним ставится специальный символ: ?-.

Если вопрос совпадает с фактом, известным системе, то ответом будет: yes или true (истина). Если системе такой факт неизвестен, то она ответит: no или false.

Как уже было сказано в предыдущем параграфе, термы в прологе могут быть составными, то есть состоять из атома-функтора, за которым идет несколько аргументов, (например: мебель(стол), родитель(X,У)). Аргументов может быть 1, 2, или даже 10, в зависимости от решаемой задачи. В качестве аргументов могут выступать названия предметов, имена, переменные или даже списки. Например, предикат “оценки” можно задать так:

| оценки(фамилия, предмет,[список_оценок]).

предикат “болезнь” может выглядеть так:

| болезнь(название_болезни, симптом1, симтом2, симптом3, симптом4).

Пролог умеет производить арифметические операции: сложение (+),вычитание (–), умножение (*),деление (/),остаток от деления целых чисел (mod), целочисленное деление (div). Некоторые версии пролога обладают боле широким набором арифметических действий.

Пример 1. Если мы хотим вычислить, к примеру, сумму чисел 2 и 5, то это будет выглядеть так: X is 2 + 5. Ответом будет: X = 7. Если же мы напишем X = 2 + 5, то ответом так и будет сумма «2 + 5».

Предикат is определен как инфиксный оператор. Его левый аргумент – или число, или неконкретизированная переменная, а правый аргумент – арифметическое выражение.

Системные предикаты =:= (равно), =\= (не равно), > (больше), < (меньше), >= (больше или равно) и <= (меньше или равно) определены как инфиксные операторы и применяются для сравнения результатов двух арифметических выражений.

Пример 2.

| маленький_рост(человек,X):-X<155.

Тогда на запрос

| ?- маленький_рост(человек,150).

мы получим ответ true (истинно), а на запрос

| ?- маленький_рост(человек,160).

ответом будет false (ложь).

Не равно можно задавать также с помощью оператора not (не): not(X=Y).

Работа со списком

Список – это последовательность данных, состоящая из конечного числа элементов. В общем случае список представляет собой абстрактный тип данных, задающий набор значений.

Обычно элементы списка заключают в квадратные скобки и перечисляют через запятую. Например: оценки([5,4,5,5,3,5])

Список следует рассматривать как структуру, состоящую из двух частей [H|T]: первый элемент, называемый головой списка; остальная часть списка, называемая хвостом. Хвост в сою очередь тоже является списком, который может быть пуст, может состоять из одного элемента, а может состоять из нескольких элементов, тогда он имеет свои собственные голову и хвост.

Пример 3. Рассмотрим список вида [а, б, в, г]. Тогда головой является элемент а, а хвостом список [б, в, г].

Элементами списка могут быть любые объекты, в том числе и списки.

Пример 4. Список науки([гуманитарные[филология,история],физико-математические[алгебра,геометрия,физика]]).

Рассмотрим примеры некоторых операций над списком.

1. Проверкапринадлежностиэлементасписку.

Пример 5. Необходимо проверить, принадлежит ли элемент б списку [а,б,в,г].

Программа в этом случае может выглядеть так:

```
принадлежит(X,[X|_]).
```

```
принадлежит(X,[_|Y]):-принадлежит(X,Y).
```

Тогда на запрос:

```
| ?-принадлежит(б,[а,б,в,г]).
```

мыполучимответ true.

2. Добавлениеэлемента в список.

Самый простой способ добавить элемент в список – это поставить элемент в начало списка, то есть сделать его головой списка.

Пример 6. Пример добавления элемента а к списку [b,c] может выглядеть так:

```
| add(X,L,[X|L]).
```

Тогда при запросе:

```
| ?-add(a,[b,c],B).
```

получим ответ:

```
| B = [a, b, c].
```

3. Удалениеэлементаизсписка.

Удаление элемента X из списка L можно задать так: удалить(X,L,L1), где X – элемент, который нужно удалить, L – список, из которого будет удален элемент, L1 – новый список, полученный из L после удаления элемента X.

Пример 7. Пример удаления элемента а из списка [a,b,c] может выглядеть так:

```
del(X,[X|T],T).
del(X,[Y|T],[Y|T1]):-del(X,T,T1).
```

Тогда на запрос:

```
|?- del(a,[a,b,c],X).
```

получим ответ:

```
| X = [b, c].
```

4. Сцеплениесписковиликонкатенация.

Эта операция позволяет произвести слияние нескольких списков в один.

Пример 8. Произведем конкатенацию списков [a,b,c] и [d,e,f].

```
конк([ ],L,L).
конк([X|L1],L2,[X|L3]):-конк(L1,L2,L3).
```

Здесь L1 и L2 – это списки, которые нужно соединить, а L3 – новый список, полученный в результате конкатенации списков L1 и L2.

Результатом запроса:

| ?- конк([a,b,c],[d,e,f],L).

будет

| L = [a, b, c, d, e, f].

5. Нахождениесуммыэлементовсписка.

Пример 9.

```
sum([ ],0).
sum([H|T],S):-sum(T,S1),S is S1+H.
```

Попробуем найти сумму элементов списка [2,3,5]:

| ?- sum([2,3,5],X).

X = 10.

6. Нахождение среднего арифметического элементов списка:

Пример 10. Программа в этом случае может выглядеть так:

```
sumn([],0,0).
sumn([H|T],N,S):-sumn(T,N1,S1),N=N1+1,S=S1+H.
avg(У,A):-sumn(У,N,S),A is S/N.
```

Здесь [ ] означает пустой список. На втором месте стоит количество элементов списка, на третьем – сумма элементов. Если список пуст, то количество его элементов и их сумма равны 0. Если же список не пуст, то отделяя по одному элементу, данная программа считает количество его элементов N и их сумму S. А затем вычисляет среднее арифметическое А. У здесь означает сам список.

7. Нахождение минимального (максимального) элемента списка:

Пример 11. Рассмотрим нахождение минимального элемента списка.

```
min_list([MinElem], MinElem):-!.
min_list([Head|Tail], MinElem):-
min_list(Tail, TailMinElem),
TailMinElem< Head, !, MinElem = TailMinElem;
```

MinElem = Head.

Знак «!» значит отсечение. Он указывает программе на завершение работы, если минимальный элемент найден.

Аналогично можно написать программу для нахождения максимального элемента списка.

2.1 Лабораторная работа № 2. Построение простой вопросно-ответной схемы

Цель работы: Научиться строить простую вопросно-ответную схему в среде Prolog.

Задания: **1.** Имеется база данных:

млекопитающее(собака).

млекопитающее(тигр).

млекопитающее(кит).

птица(воробей).

птица(канарейка).

птица(синица).

птица(ласточка).

рыба(карась).

рыба(скат).

рыба(камбала).

1. Определить, является ли рыбой а) скат; б) кит.

2. Определить, является ли птицей а) воробей; б) ворона.

2. Задача выбора должности для соискателя, желающего получить работу в компании.

Если у соискателя нет диплома, то соискателю отказать в приеме на работу.

Если у соискателя есть диплом, то он, возможно, будет принят на работу.

Если соискатель имеет диплом и он сделал важное открытие, то предложить соискателю должность научного сотрудника.

Если средний балл за время обучения >=3.5, то предложить соискателю должность инженера.

Если средний балл <3.5 и соскатель работал по специальности >3 лет, то предложить соискателю должность инженера по безопасности.

Если средний балл <3.5 лет и стаж работы <2 лет, то соскателю отказать.

3. Что делать, если Вы заболели?

Если у Вас грипп и Вы находитесь в возрасте риска, то вызовите доктора.

Если у Вас ларингит, то вызовите доктора.

Если у Вас простуда, то примите лекарство и ложитесь в постель.

Если у вас ангина, то вызовите врача.

Если у Вас грипп и вы не в возрасте риска, то примите антигриппин и ложитесь в постель.

Если у Вас высокая температура и болят мышцы, то это грипп.

Если у Вас насморк, болят мышцы и нет температуры, то это простуда.

Если у Вас болит горло, лающий кашель и высокая температура, то это ларингит.

Если у вас болит горло, высокая температура и есть налеты в горле, но нет кашля, то у вас ангина.

Если Вам меньше 7 или больше 65 лет, то Вы находитесь в возрасте риска.

4. Сведения об успеваемости учеников представлены предикатами вида:оценка(фамилия, предмет , список_оценок).

1.Определить предикат не_успевает ученик X по предмету P , если у него есть двойка по этому предмету.

2. Определить предикат двоечник, если ученик не успевает по какому-либо предмету.

3. Определить предикат отличник для ученика X по предмету P, если средний балл по этому предмету близок к 5.

Вопросы для собеседования при защите работы:

1. Интеллектуальныйинтерфейс.

2. Классификацияуровнейпонимания.

3 Методы решения задач

Одной из важнейших частей интеллектуальных систем является теория решения задач.

При решении задачи в любой интеллектуальной системе должен быть построен план действий.

Все существующие методы планирования решения задач делят на два вида: планирование в пространстве состояний (называют SS-проблемой) и планирование в пространстве задач (называют PR-проблемой).

Планирование в пространстве состояний представляет собой нахождение пути решения от исходного состояния к конечному.

Пространство состояний удобно представлять в виде графа, вершины которого – это состояния или ситуации в процессе решения задачи, а дуги – шаги перехода из одних состояний в другие. Причем, особо выделяют множество начальных состояний и множество конечных (целевых) состояний.

Решение задачи сводится к нахождению оптимального пути на графе.

К задачам, решаемым с помощью планирования в пространстве состояний, относят:

- задача о восьмиферзях;
- головоломка «игра в восемь»;
- задача о «ханойскойбашне»;
- переупорядочение кубиков, поставленных друг на друга в виде столбиков;
- задача о волке, козе и капусте;
- задача о двухкувшинах;
- задача о коммивояжере;
- другиеоптимизационныезадачи.

Выделяют слепые и направленные методы поиска. Слепой метод имеет два вида:

- поиск вглубь (рисунок 1): в этом случае каждая альтернатива исследуется до конца, без учета остальных альтернатив. То есть каждый раз выбирается самая “глубокая” вершина, та, которая расположена дальше других от

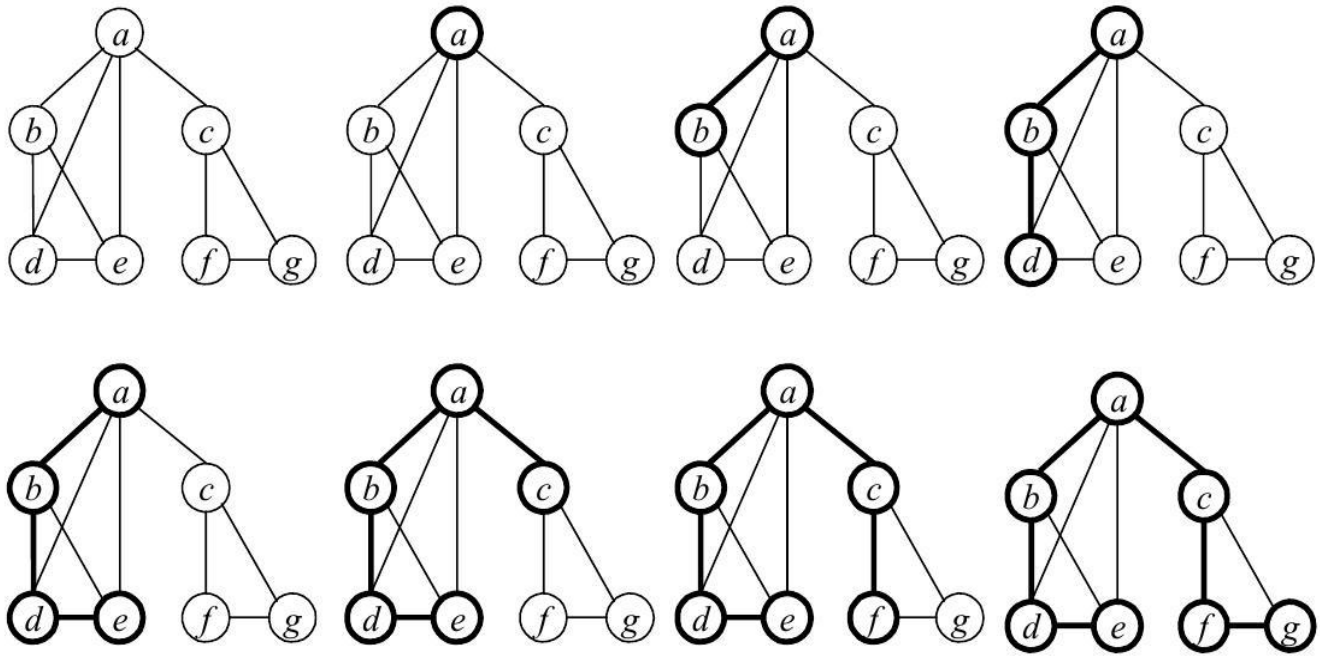


Рисунок 1 –Поисквглубь

начальной вершины. Метод не подходит для "высоких" деревьев, так как легко можно пропустить нужную ветвь и затратить много усилий на исследование альтернатив, не принципиальных для решения задачи.

- поиск вширь (рисунок 2): на каждом уровне пространства состояний анализируются все возможные альтернативы и только после этого осуществляется переход на следующий уровень. Метод может оказаться хуже метода поиска вглубь,

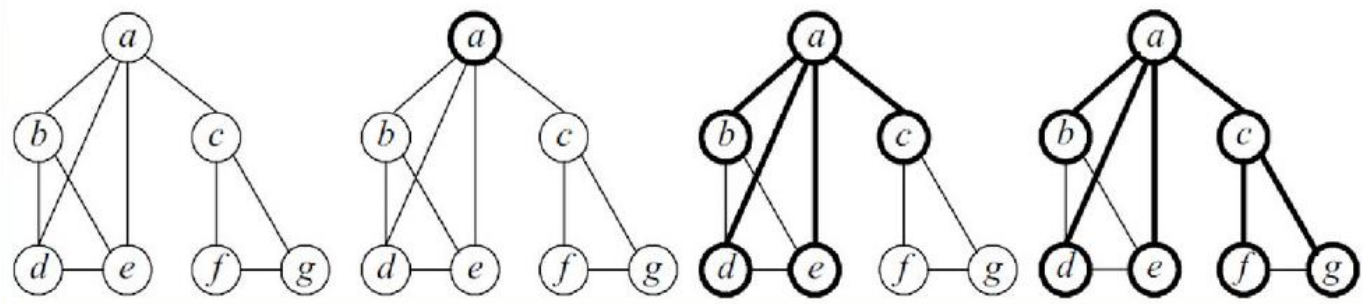


Рисунок 2 –Поисквширь

если в графе все пути, ведущие к целевой вершине, расположены примерно на одной и той же глубине.

Поиск в глубину наиболее соответствует рекурсивному стилю программирования, принятому в Прологе.

Оба слепых метода требуют большой затраты времени и поэтому необходимы направленные методы поиска.

К этим методам относят следующие алгоритмы:

- методветвей и границ;
- алгоритмкратчайшихпутейМура;
- алгоритмДейкстры;
- алгоритм Дорана и Мичи поиска с низкой стоимостью;
- алгоритм Харта, Нильсона и Рафаэля.

В противоположность слепым методам, направленные методы используют априорную (эвристическую) информацию о местоположении цели в пространстве состояний, поэтому для раскрытия обычно выбирается более перспективная вершина. Это делается для уменьшения возникающего перебора.

По направлению поиска алгоритмы делят на следующие виды:

- прямые: поиск ведется от начальной вершины к конечной (целевой), как правило, используется, когда конечное (целевое) состояние задано неявно;
- обратные: поиск от целевой вершины в направлении к начальной, и используется тогда, когда исходное состояние задано неявно, а целевое явно;
- двунаправленные: происходит чередование прямого и обратного поиска, или же одновременное их проведение. Необходимо отметить, что такой метод требует удовлетворительного решения двух проблем: смены направления поиска и оптимизации «точки встречи». Одним из критериев для решения первой проблемы является сравнение «ширины» поиска в обоих направлениях – выбирается то направление, которое сужает поиск. Вторая проблема вызвана тем, что прямой и обратный пути могут разойтись и чем уже поиск, тем это более вероятно[5].

Наиболее часто используемыми являются прямые алгоритмы, так как являются наиболее простыми.

Планирование в пространстве задач состоит в последовательном разбиении исходной задачи наболее простые до тех пор, пока не будут получены только элементарные задачи. То есть пространство образуется в результате введения на множестве задач отношения типа: "часть – целое", "задача – подзадача", "общий случай – частный случай" и т.п. PR-проблема состоит в выделении подзадач, решение которых системе известно.Если правильно понять суть задачи и разделить ее на иерархически связанные подзадачи, то можно добиться минимального пути поиска к ее решению. Не обязательно требовать, чтобы основная задача и все ее подзадачи решались одинаковыми методами.

При разбиении задачи на множество подзадач удобно использовать И/ИЛИ-граф. В этом графе каждая вершина, кроме концевой, имеет либо конъюнктивно связанные дочерние вершины (И-вершина), либо дизъюнктивно связанные (ИЛИ-вершина). В частном случае, при отсутствии И-вершин, имеет место граф пространства состояний. Концевые вершины являются либо заключительными (им соответствуют элементарные задачи), либо тупиковыми. Начальная вершина (корень И/ИЛИ-графа) представляет исходную задачу. Цель поиска на И/ИЛИ-графе-показать, что начальная вершина разрешима. Разрешимыми являются заключительные вершины (И-вершины), у которых разрешимы все дочерние вершины, и ИЛИ-вершины, у которых разрешима хотя бы одна дочерняя вершина. Разрешающий граф состоит из разрешимых вершин и указывает способ разрешимости начальной вершины. Наличие тупиковых вершин приводит к неразрешимым вершинам. Неразрешимыми являются тупиковые вершины, И-вершины, у которых неразрешима хотя бы одна дочерняя вершина, и ИЛИ-вершины, у которых неразрешима каждая дочерняя вершина [5].

К этому методу планирования относят:

- алгоритмЧенга и Слейгла;
- методключевыхоператоров.

Рассмотрим реализацию алгоритма поиска в глубину на языке Prolog.

Пример 1. Пусть дан неориентированный граф (рисунок 3). Требуется найти все пути из вершины a в вершину c.

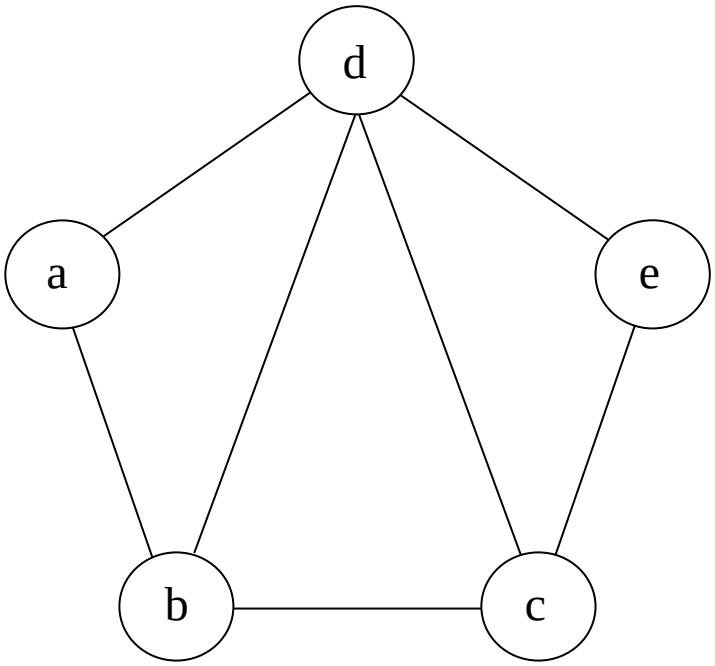


Рисунок 3 –Неориентированныйграф

Сначала необходимо указать все связанные вершины. Поскольку граф неориентированный, то путь из вершины ав b и из b в a – это одно и то же.

```
ш(a,b).
ш(b,c).
ш(a,d).
ш(b,d).
ш(c,d).
ш(c,e).
ш(d,e).
шаг(A,B):-ш(A,B);ш(B,A).
продлить([H|T],[New,H|T]):-шаг(H,New),not(member(New,[H|T])).
путь([Finish|T],Finish,[Finish|T]).      %если данная вершина совпадает с
                целевой, то путь найден
путь(TempWay,Finish,Way):-
```

```
продлить(TempWay,NewWay), %делаемшаг
путь(NewWay,Finish,Way).%продолжаем поиск пути
с учетом сделанного шага
искать_путь(Start,Finish):-
   путь([Start],Finish,Way),%вызываем поиск в глубину,
считая, что пока путь состоит
только из начальной вершины
показать_ответ(Way).%выводим путь на экран в наглядном
виде
показать_ответ([_]):-!.
показать_ответ([A,B|Tail]):-
показать_ответ([B|Tail]),nl,write(B),write(' -> '),write(A).
```

Задаем запрос:

```
| ?-искать_путь(a, c).
```

Система выдаст все варианты пути из вершины a в вершину c:

```
a -> b
b -> c
true ;

a -> b
b -> d
d -> e
e -> c
true ;

a -> b
b -> d
d -> c
true ;

a -> d
```

```
d -> e
e -> c
true ;

a -> d
d -> b
b -> c
true ;

a -> d
d -> c
true ;
false.
```

С помощью языка Пролог можно решать логические задачи, чего нельзя сделать с помощью большинства процедурных языков.

Большинство таких задач требуют установления соответствия между двумя или несколькими множествами с одинаковым количеством элементов.

На Прологеданные множества можно описывать как базы данных, а зависимости между объектами устанавливать с помощью правил.

Пример 2. В соревнованиях на велосипедах три первых места заняли Андрей, Павлик и Костя. Какое место занял каждый из них, если Павлик занял не второе и не третье место, а Костя – не третье?

Имя	I место	II место	III место
Андрей			
Павлик		–	–
Костя			–

Обычно такие задачи решаются с помощью таблицы.По условию Павлик занял не второе и не третье место, а Костя – не третье. Значит, необходимо поставить символ ‘–’ в соответствующих ячейках таблицы.

Между множеством имен участников соревнований и множеством мест должно необходимо установить взаимно-однозначное соответствие.

То, что Павлик занял не второе и не третье место позволяет поставить знак ‘+’ в ячейку с первым местом. Значит, автоматически можно поставить ‘–’ в ячейки с первым местом у Андрея и Кости. В строке Кости остается только одна пустая ячейка – это второе место, ставим там знак ‘+’,значит, в ячейке 2 места Андрея ставим ‘–’. Остается только одна пустая ячейка у Андрея, ставим в нее знак ‘+’.

Итоговаятаблицавыглядиттак:

Имя	I место	II место	III место
Андрей	–	–	+
Павлик	+	–	–
Костя	–	+	–

Из этой таблицы видно, что Андрей занял III место, Павлик – I место, Костя – II место.

Рассмотрим, как будет выглядеть программная реализация данной задачи на языке Пролог.

Сначала рассматриваем два множества данных: имена и места. Затем устанавливаются правила, задающие соответствия между этими множествами.

```
имя(андрей).
имя(павлик).
имя(костя).
место(первое).
место(второе).
место(третье).
соответствие(X, Y) :- имя(X),  X=павлик,
        место(Y), not(Y=второе), not(Y=третье).
соответствие(X, Y) :- имя(X), X=костя,  место(Y), not(Y=третье).
```

```
соответствие(X, Y) :- имя(X),  X=андрей, место(Y).
решение(X1,Y1,X2,Y2,X3,Y3) :-
        X1=андрей, соответствие(X1,Y1),
        X2=павлик, соответствие(X2,Y2),
        X3=костя, соответствие(X3,Y3),
        Y1\=Y2, Y2\=Y3, Y1\=Y3.
```

Строки 1-3 задают множество имен, а строки 4-6 – множество мест.

Следующие 5 строк устанавливают взаимно-однозначное соответствие между элементами X и У данных множеств, где X– элемент из множества имен, У– элемент из множества занятых мест. Здесь же указываются данные из условия.

Так как увсех ребят разные места, то это необходимо прописать в коде Y1\=Y2, Y2\=Y3, Y1\=Y3.

На запрос:

```
| ?- решение(X1,Y1,X2,Y2,X3,Y3).
```

системавыдаст:

```
X1 = андрей,
Y1 = третье,
X2 = павлик,
Y2 = первое,
X3 = костя,
Y3 = второе ;
false.
```

Так как при запросе других решение (при нажатии ;), получаем false, то это значит, что других решений у системы нет.

Пример 3. Валя, Юля и Маша встали рядом. Расположить девочек в порядке убывания роста, если известно, что Маша выше Юли, а Валявыше Маши.

В условии задачи перечисляются объекты одного типа, связанные между собой соотношением «выше».

Реализация данной задачи на Прологе будет выглядеть так:

```
выше(юля,маша).
выше(маша,валя).
ряд(X,Y,Z):-выше(Y,X),выше(Z,Y).
```

Сделаем запрос:

| ?- ряд(X, Y, Z).

Получимрешениезадачи:

```
X = валя,
Y = маша,
Z = юля.
```

3.1 Лабораторная работа № 3. Решение логических задач

Цель работы: Научиться решать логические задачи среде Prolog.

Задания:

1. Лабиринт представляет собой систему комнат, соединенных между собой переходами. В лабиринте имеется вход и выход, а также комната с золотым кладом. Кроме того, имеются комнаты, запрещенные для посещений: комната монстров и комната разбойников.

а) Найди путь в лабиринте от входа до входа, не посещая дважды одной и той же комнаты;

б) найти путь с посещением золотой комнаты;

в) найти путь, избегающий запрещенных к посещению комнат.

2. ***Задача о волке, козе и капусте.*** У фермера есть волк, коза и капуста. Все они находятся на левом берегу реки. Необходимо перевезти это «трио» на правый берег, но в лодку может поместиться что-то одно – волк, коза или капуста. Нельзя оставлять на одном берегу волка с козой и козу с капустой.

3. Коля, Боря, Вова и Юра заняли первые четыре места в соревновании. На вопрос, какие места они заняли, трое из них ответили:

1) Коля ни первое, ни четвертое;

2) Борявторое;

3) Вованебылпоследним.

Какое место занял каждый мальчик?

4. Ваня, Петя,Саша и Коля носят фамилии, начинающиеся на буквы В, П, С, К. Известно, что: 1) Ваня и С. – отличники; 2) Петя и В. – троечники; 3) В.– ростом выше П.; 4) Коля ростом ниже П.; 5) Саша и Петя имеют одинаковый рост.

На какую букву начинается фамилия каждого мальчика?

5. Катя, Оля и Маша вышли на прогулку, причем туфли и платье каждой были или красного, или синего, или белого цвета. У Кати были белые туфли, а Оля не любит белый цвет. Только у Маши платье и туфли были одного цвета. Определить цвет туфель и платья каждой из девочек, если у всех туфли и платья были разного цвета.

6. Беседуют трое друзей: Белокуров, Чернов и Рыжов. Один из них брюнет, другой блондин, третий – рыжий. Брюнет обратил внимание Белокурова на тот факт, что ни у кого из троих цвет волос не соответствует фамилии. Какойцветволос у каждогоиздрузей?

Вопросы для собеседования при защите работы:

1. Методырешениязадач.
2. Решение задач методом поиска в пространстве состояний.
3. Решениезадачметодомредукции.
4. Решениезадачдедуктивноговыбора.
5. Решение задач, использующие немонотонные логики, вероятностные логики.

4 Представление знаний в интеллектуальной системе

В направлении «Представление знаний» решаются задачи, связанные с формализацией и представлением знаний в интеллектуальных системах. Для этого разрабатываются специальные модели представления знаний и языки для описания знаний, выделяются различные типы знаний. Изучаются источники, из которых ИС может черпать знания, и создаются процедуры и приемы, с помощью которых возможно приобретение знаний для ИС. Проблема представления знаний для ИС чрезвычайно актуальна, т.к. ИС – это система, функционирование которой опирается на знания о проблемной области, которые хранятся в ее памяти[5].

На данное время нет точного и четкого определения понятия «знание».

Обычно под знанием понимают результат познавательной деятельности человека, направленный на отражение тех или иных свойств существующей действительности.

Выделяют различные виды знаний: научное, обыденное, религиозное, интуитивное и др.

Представление знаний – это отражение знаний на естественном или формальном языке.

В естественных и технических науках традиционным считается следующий способ представления знаний: сначала на естественном языке вводятся основные понятия и отношения между ними, которые затем могут использоваться для определения последующих понятий и отношений; далее, устанавливается соответствие между какими-либо характеристиками понятий и подходящей математической моделью. Основная цель данного представления знаний – строить математические модели реального мира и его частей, для которых соответствие между системой знаний может быть установлено на основе совпадения имен переменных модели и имен понятий без предварительных пояснений и установления дополнительных соответствий.

Считается, что универсальным языком представления знаний, является естественный язык, но его использование в информационных системах

затруднительно, так как не существует алгоритмов для обработки естественного языка.

Данные – это некоторая зарегистрированная информация об объектах, явлениях окружающей среды, предназначенная для обработки, интерпретации, передачи человеком или информационными системами.

Рассмотрим некоторые свойства знаний, описанные в [5]:

1. Внутренняяинтерпретируемость.

В память машины вводится информация о некоторой протоструктуре информационных единиц, т. е. специальное машинное слово, в котором указано, в каких разрядах хранятся данные. Атрибуты могут играть роль имен для машинных слов, которые соответствуют строкам таблицы. Каждая строка является экземпляром протоструктуры. Свойство внутренней интерпретируемости всех информационных единиц обеспечивается СУМД (система управления манипулирования данными), хранимой в реляционной БД. Это позволяет получать от системы ответы о содержимом, хранящемся в ее памяти.

2. Структурированность.

Данные должны быть структурированы. Каждая информационная единица может входить в состав другой информационной единицы или содержать в себе более примитивные информационные единицы. Это позволяет вводить на множестве информационных единиц отношения типа «целое-часть», «множество-элемент» и т.п.

3. Связность.

Между информационными единицами должны устанавливаться связи различного типа. Они могут иметь тип «причина-следствие» (каузальные отношения), могут быть функциональными, структуризации, пространственно-временными, определять порядок выбора информационных единиц и их совместимость.

4. Семантическаяметрика.

Очень полезно задавать между информационными единицами отношения, описывающие их ситуационную близость. Такие отношения помогают находить

знания, близкие к уже имеющимся, описывать ситуации, происходящие по схожим сценариям.

5. Шкалирование.

Между информационными единицами можно задавать количественные соотношения (в этом случае речь идет о метрических шкалах) и отношения порядка (порядковые шкалы). Количественные соотношения могут быть выражены лингвистически, например, никогда – 0, всегда – 1, остальным значениям «очень редко», «редко», «часто» и т.п. на шкале будут соответствовать интервалы. В этом случае можно говорить о нечетких метрических шкалах. Если же эти отношения выражены с помощью пар-антонимов, например, черный-белый и т.п., то шкалы называют оппозиционными.

6. Активность.

Обычно в информационных системах данные (декларативные знания), хранящиеся в базе данных, активизируются с помощью программ (процедурные знания), работающих с этими данными. В интеллектуальных системах несколько иная ситуация. Знания могут активизировать работу программы. Появление в базе фактов или описаний событий, установление связей может стать источником активности системы.

Перечисленные особенности информационных единиц определяют ту грань, за которой данные превращаются в знания, а базы данных перерастают в базы знаний (БЗ).

Совокупность средств, обеспечивающих работу со знаниями, образуетсистему управления базой знаний (СУБЗ).

4.1 Лабораторная работа №4. Методы извлечения знаний

Цель работы: Получить практический опыт использования методов извлечения и представления знаний, используемых при проектировании интеллектуальных систем.

Задания:

1. Генерация идей методом мозгового штурма.

Идея должна удовлетворять следующим условиям:

- относиться к робототехнике (аппаратное устройство с программным управлением);
- является функционально доступным большинству людей;
- доступно по стоимости для конечного пользователя.

2. Созданиепаспортаидеи:

а) брендирование и разработка концепции (придумать название, слоган и логотип);

б) исследование пользовательской аудитории (выделить одну или несколько пользовательских групп);

в) описаниефункционала.

3. Составление интеллект-карты идеи (типы устройств, пользователи, основные и дополнительные функции и др.).

Вопросы для собеседования при защите работы:

1. Данные и знания. Основные понятия инженерии знаний.

2. Переход от Базы Данных к Базе Знаний.

3. Перечислите основные компоненты базы знаний СИИ.

4. Перечислите основные этапы представления знаний при обработке на компьютере.

5. Перечислитеосновныевидызнаний.

6. Чтотакоепроцедурныезнания?

7. Чтотакоепредставлениезнаний?

5 Модели представления знаний

Выделяют 4 класса моделей представления знаний [5], [10].

1. *Формальные логические модели.* В основе моделей данного типа лежит формальная система, задаваемая четверкой вида: $M = <T, P, A, B>$.

Множество*T*есть множество базовых элементовразличной природы, например, множество деталей детской мозаики, множество слов некоторого ограниченного словаря, входящих в состав некоторого набора и т.п. Важно, что для множества *T* имеется способ, позволяющий определить, принадлежитнекоторый элемент этому множеству или не принадлежит. Правило данной проверки может быть произвольным, главное, что за конечное число шагов, мы должны получить ответ, является ли *x* элементом множества *T*. Обозначим эту процедуру П(*T*).

Слова, описывающие сущности предметной области, называются *термами*.

Множество*P* – это множество синтаксических правил. С помощью этих правил из элементов множества *T*строятсинтаксически правильные совокупности. Например, из деталей детской мозаикисобираются новые картинки, из слов ограниченного словаря строят фразы или предложения. Существует процедураП(*P*), с помощью которой за конечное число шагов можно получить ответ на вопрос, является ли совокупность *X* синтаксически правильной.

В множестве синтаксически правильных совокупностей выделяется некоторое подмножество *A*, элементы которого называются аксиомами. В соответствии с этим, так же как и для других множеств, существует процедура П(*A*), с помощью которой для любой синтаксически правильной совокупности можно получить ответ на вопрос принадлежит ли она множеству *A*.

Множество *B* является множеством правил вывода. Применяя данные правила к элементам множества*A*, можно получать новые синтаксически правильные совокупности. К полученным синтаксически правильным совокупностям снова можно применять правила вывода из*B*.

Например, если из формул F_1, F_2,..., F_n непосредственно выводится формула *F*, то это можно записать в виде

$$\frac{F_1, F_2, \ldots, F_n}{F},$$

гдеF_1, F_2,..., F_n называются посылками правила, а F – его следствием (заключением).

Таким образом, формируется множество выводимыхв данной формальной системе совокупностей. Если существует процедура П(B), с помощью которой можно определить для любой синтаксически правильной совокупности, является ли она выводимой, то соответствующая формальная система называется разрешимой. Это показывает, что именно правило вывода является наиболее сложной составляющей формальной системы.

По сути, формальная система представляет собой генератор порождения новых знаний, образующих множествовыводимыхв данной системезнаний. МножествоA образуют знания, введенные извне, а с помощью множества B, образуются новые знания внутри системы. Благодаря этому свойству, логические модели широко используются в базах знаний. Оно позволяет хранить в базе лишь те знания, которые образуют множество A, а все остальные знания получать из них с помощью правил вывода.

Примером логических моделей может служить логика предикатов.

Пример 1. Рассмотрим пример из [10]. Введем обозначения:

$A(x)$ = «студент x учится на отлично»;

$B(x)$ = «студент x получает повышенную стипендию».

Зададим формулу:

$$A(\text{Иванов}) \rightarrow B(\text{Иванов}).$$

Данная формула означает, «если студент Иванов учится на отлично, то студент Иванов получает повышенную стипендию».

Если применить квантор всеобщности, то формула примет вид:

$$(\forall x)(A(x) \rightarrow B(x))$$

и будет означать «каждый студент, который учится на отлично, получает повышенную стипендию».

Рассмотрим подробнее, на чем основан логический вывод.

Действие компоненты вывода в логических моделях основано на применении следующих правил вывода [10]:

1) *ModusPonens* “если верноA и верно, что из A следует B, то верно B”:

$$\frac{A \to B, A}{B}.$$

2) *ModusTollens* “если известно, что изA следует B и B ложно, то и A ложно”:

$$\frac{A \to B, \neg B}{\neg A}.$$

3) *ModusPonendoTollens* “если верно илиA, или B, и верно одно из них, то другое ложно”:

$$\frac{A \oplus B, A}{\neg B}, \frac{A \oplus B, B}{\neg A}.$$

4) *ModusTollendoPonens*

а) “если верноA или B (в разделительном смысле) и ложно одно из них, то верно другое”:

$$\frac{A \oplus B, \neg A}{B}, \frac{A \oplus B, \neg B}{A}.$$

б) “если верноA или B (в неразделительном смысле) и ложно одно из них, то верно другое”:

$$\frac{A \vee B, \neg A}{B}, \frac{A \vee B, \neg B}{A}.$$

5) Правило транзитивности “если изA следует B, а из B следует C, то из A следует C”:

$$\frac{A \to B, B \to C}{A \to C}.$$

6) Закон противоречия “если изA следует B и $\neg B$, то ложно A”:

$$\frac{A \to B, A \to \neg B}{\neg A}.$$

7) Правило контрапозиции (или метод резолюций) “если из A следует B, то из того, что ложно B, следует, что ложно A”:

$$\frac{A \to B,}{\neg B \to \neg A}.$$

2. *Семантические сети*. Данные модели формально можно задать в виде$H = <I, C_1, C_2, ..., C_n, \Gamma>$. Здесь$I$есть множество информационных единиц или множество сущностей (объектов);$C_1, C_2, ..., C_n$ - множество типов связей между данными информационными единицами. Отображение Г задает между информационными единицами, входящими вI, связи из заданного набора типов связей.

Термин семантическая значит смысловая.

В зависимости от типов связей, используемых в модели, различают*классифицирующие сети, функциональные сети*и*сценарии*. В классифицирующих сетях используются отношения структуризации. Такие сети позволяют в базах знаний вводить разные иерархические отношения между информационными единицами. Функциональные сети характеризуются наличием функциональных отношений. Их часто называют *вычислительными моделями*, т.к. они позволяют описывать процедуры "вычислений" одних информационных единиц через другие. В сценариях используются каузальные отношения, а также отношения типов "средство – результат", "орудие – действие" и т.п. [5].

Семантическую сеть удобно представлять в виде ориентированного графа, вершины которого – это сущности (информационные единицы), а дуги – отношения между ними.

Пример 2. Рассмотрим пример семантической сети из [11] (рисунок 4).

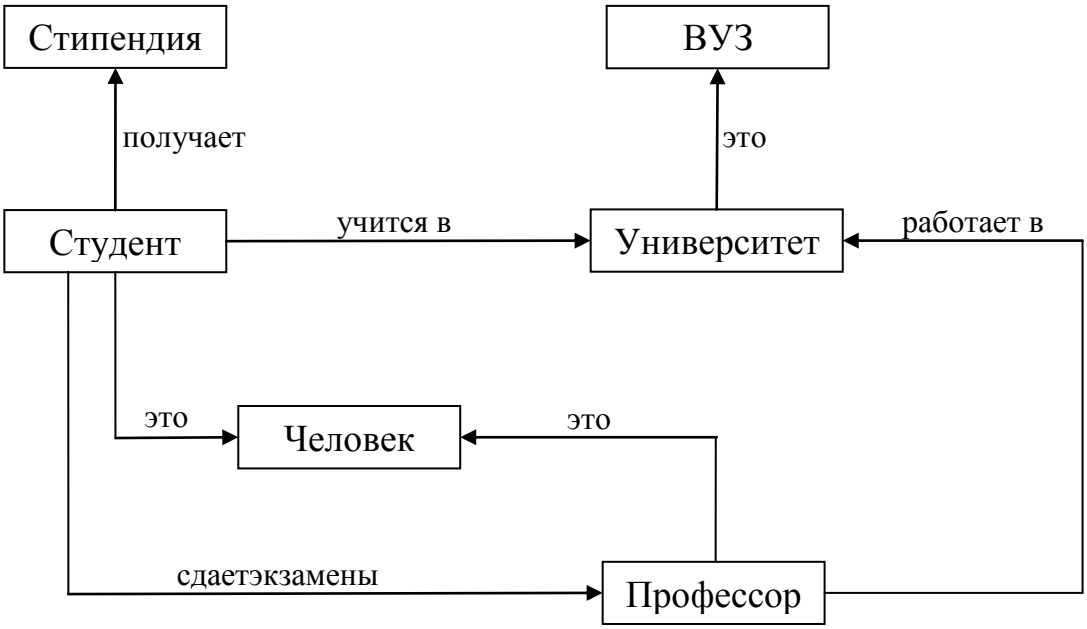


Рисунок4 –Семантическаясеть

Основное преимущество семантических сетей – наглядность представления знаний.

Недостатками данной модели являются сложность систематизации отношений, сложность поиска вывода, сложность пополнения и удаления из сети новых знаний.

3. *Продукционные модели*. В моделях этого типа используются некоторые элементы логических и сетевых моделей. Из логических моделей заимствована идея правил вывода, которые здесь называются продукциями, а из сетевых моделей – описание знаний в виде семантической сети. В результате применения правил вывода к фрагментам сетевого описания происходит трансформация семантической сети за счет смены ее фрагментов, наращивания сети и исключения из нее ненужных фрагментов. Таким образом, в продукционных моделях процедурная информация явно выделена и описывается иными средствами, чем декларативная информация. Вместо логического вывода, характерного для логических моделей, в продукционных моделях появляется вывод на знаниях[5].

Общая схема продукционной системы представлена на рисунке 5 [11].

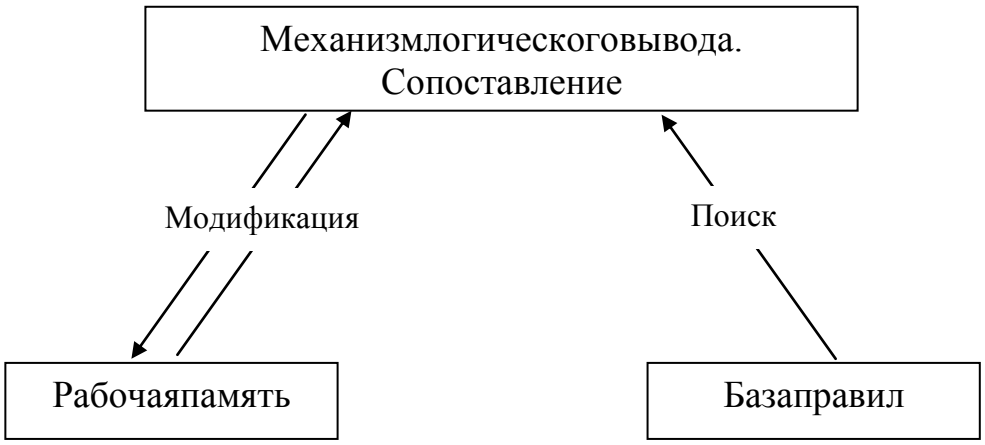


Рисунок5 –Схемапродукционнойсистемы

Знания в данной модели представлены в виде: ЕСЛИ (условие), ТО (действие). По “условию” (в качестве которого выступает элементарное предложение или несколько предложений, связанных логическими связками) осуществляется поиск в базе знаний и при успешном исходе поиска выполняется “действие”. “Условие” называют антецедентом, “действие” – консеквентом. Поэтому процедуру вывода в продукционных системах называют «вывод на знаниях».

Пример 3. а) ЕСЛИ «идет дождь», ТО «нужно взять зонт».

б) ЕСЛИ «болит горло» И «высокая температура», ТО «болезнь – ангина».

Существует два типа продукционных систем: 1) с прямым выводом; 2) с обратным выводом. Для прямого вывода характерно направление от фактов к заключению, для обратного – от гипотез вероятных заключений, которые либо подтверждаются, либо нет фактами, поступающими в рабочую память. Существуют системы с двунаправленным выводом[10].

Данная модель чаще всего используется в экспертных системах. Главными ее достоинствами являются наглядность, высокая модульность, легкость для внесения изменений и простой логический вывод.

4. *Фреймовые модели*. Фреймовая модель представления знаний основана на теории фреймов М. Минского, которая представляет собой психологическую модель памяти человека и его сознания[10].

Во фреймовых моделях,в отличие от моделей других типов, фиксируется жесткая структура информационных единиц, которая называется протофреймом.

Фрейм – это структура, состоящая из имени фрейма, слотов и присоединенных процедур, связанных с фреймом или слотами.

В общем виде структура фрейма выглядит следующим образом:

Имя фрейма:

Имя слота 1(значение слота 1)

Имя слота 2(значение слота 2)

. . . . . .

Имя слота К (значение слота К).

Имя служит для идентификации описываемого понятия.

Значением слота могут быть числа или математические соотношения, тексты на естественном языке или программы, правила вывода или ссылки на другие слоты данного фрейма или других фреймов и даже набор слотов более низкого уровня, что позволяет во фреймовых представлениях реализовать "принцип матрешки".

При конкретизации фрейма ему и слотам присваиваются конкретные имена и происходит заполнение слотов. Таким образом, из протофреймов получаются*фреймы – экземпляры.* Переход от исходного протофрейма к фрейму – экземпляру может быть многошаговым, за счет постепенного уточнения значений слотов.

Связи между фреймами задаются значениями специального слота с именем "Связь".

Пример 4. Рассмотрим пример фрейма «Научная конференция».

Имяслота	Значениеслота	IF-NEDDED
Дата	15.04.18	
Местопроведения	Аудитория 1605	
Темадоклада	Системыестественно-языковогообщения	
Докладчик	Иванов И.И.	Кто?

Запрос «Кто?» автоматически запускается при обращении к слоту «Докладчик». Если значение этого слота не определено, то происходит генерация

запроса «Кто выступает?» и полученный ответ записывается в качестве значения слота.

В интеллектуальных системах с фреймовым представлением знаний используется три способа управления логическим выводом: демоны, присоединенные процедуры и механизм наследования, который является и единственным для фреймовых объектно-ориентированных систем. Управленческие функции механизма наследования заключаются в автоматическом поиске и определении значений слотов фреймов нижележащих уровней по значениям слотов фреймов верхних уровней, а также в запуске присоединенных процедур и демонов. Присоединенные процедуры и демоны могут реализовать любой механизм вывода, однако, эта реализация имеет конкретный характер и требует значительных затрат труда проектировщиков и программистов[10].

Основной механизм вывода из фреймовой модели – сопоставление (matching).

Различают синтаксическое сопоставление, когда сравнивается структура единиц знания (фреймов или слотов), и семантическое сопоставление, когда сравнивается содержимое этих единиц. Результат сопоставления может быть бинарным или иметь параметрический характер (введенный параметр отражает степень сопоставимости образцов) [5].

Часть специалистов по ИС считает, что нет необходимости специально выделять фреймовые модели в представлении знаний, т.к. в них объединены все основные особенности моделей остальных типов.

Пример 5.

(Предметная область – Медицина)

Переменные:

1) СГ – состояниегорла;
2) Н – насморк;
3) К – кашель;
4) СМ – состояниеминдалин;
5) Г – голос;
6) Т – температура;

7) ГБ – головнаяболь;

8) НС – наличиесыпи;

9) ОИ – общаяинтоксикация;

10) ДЛ – давление;

11) Д – диагноз.

Правила:

1) ЕСЛИ СГ = *красное* ТО К = *сухой*;

2) ЕСЛИ К = *сухой* И Т = *невысокая* И Н = *да* ТО Д = *ОРВ*;

3) ЕСЛИ ОИ = *тяжелая* И СМ = *воспаленные* И ГБ = *да* ТО Д = *ангина*;

4) ЕСЛИ ДЛ = *высокое* И СГ = *красное* И СМ = *фиброзная пленка* И ГБ = *да* ТО Д = *дифтерия*;

5) ЕСЛИ Т = *высокая* И Н = *да* И К = *сухой* И НС = *да* ТО Д = *краснуха*;

6) ЕСЛИ ОИ = *тяжелая* Г = *охрипший* И НС = да ТО Д = *корь*;

7) ЕСЛИ Т = *высокая* ТО ОИ = *тяжелая*;

8) ЕСЛИ ОИ = *тяжелая* И НС = *да* ТО Д = *оспа*;

9) ЕСЛИ К = *влажный* И ОИ = *тяжелая* ТО Д = *бронхит*;

10) ЕСЛИ К = *сухой* ТО Г = *охрипший*;

11) ЕСЛИ ОИ = *тяжелая* ТО ДЛ = *высокое*.

5.1 Лабораторная работа № 5. Представление знаний в виде правил

Цель работы: Изучение основных приемов представления знаний в виде правил продукции.

Задания: 1. Реализовать продукционную систему, помогающую в постановке диагноза на основании правил из примера, указанного выше. Для реализации использовать среду Prolog.

2. Построить набор продукционных правил для решения задачи: Требуется найти такую последовательность ходов конем, при которой он ставится на каждую клетку квадратной доски размером 3×3 только один раз.

1	**2**	**3**
4	**5**	**6**
7	**8**	**9**

3. Построить набор продукционных правил для определения вида одежды в зависимости от погодных условий. Реализовать данную модель в среде Prolog.

Вопросы для собеседования при защите работы:

1. Перечислите основные виды моделей представления знаний.
2. Дайте характеристику формальным логическим моделям представления знаний.
3. Чтотакоетерм?
4. На чем основано представление знаний с помощью семантической сети?
5. Как фреймовые системы могут использоваться для представления знаний?
6. В чем состоит представление знаний с помощью продукций?
7. Как осуществляется вывод на знаниях в продукционной модели?
8. Какие способы управления логическим выводом и интеллектуальных системах с фреймовым представлением вы знаете?
9. Приведите пример представления знаний с помощью фреймовой модели.
10. На чем основан вывод решения в логических моделях представления знаний?
11. Методы вывода решения в логических моделях представления знаний.

6 Экспертные системы

Экспертная система–компьютерная программа, способная заменить специалиста–эксперта в разрешении проблемной ситуации.

Структура экспертных систем.

Типичная статическая ЭС состоит из следующих основных компонентов (рисунок 6):

- решателя (интерпретатора);
- рабочей памяти (РП), называемой также базой данных (БД);
- базызнаний (БЗ);
- компонентовприобретениязнаний;
- объяснительногокомпонента;
- диалоговогокомпонента.

*База данных (рабочая память)*предназначена для хранения исходных и промежуточных данных решаемой в текущий момент задачи. Этот термин совпадает по названию, но не по смыслу с термином, используемым в информационно-поисковых системах (ИПС) и системах управления базами данных (СУБД) для обозначения всех данных (в первую очередь долгосрочных), хранимых в системе.

База знаний(БЗ) в ЭС предназначена для хранения долгосрочных данных, описывающих рассматриваемую область (а не текущих данных), и правил, описывающих целесообразные преобразования данных этой области.

Решатель, используя исходные данные из рабочей памяти и знания из БЗ, формирует такую последовательность правил, которые, будучи примененными к исходным данным, приводят к решению задачи.

*Компонент*приобретения знаний автоматизирует процесс наполнения ЭС знаниями, осуществляемый пользователем – экспертом.

*Объяснительный компонент*объясняет, как система получила решение задачи (или почему она не получила решение) и какие знания она при этом использовала, что облегчает эксперту тестирование системы и повышает доверие пользователя к полученному результату.

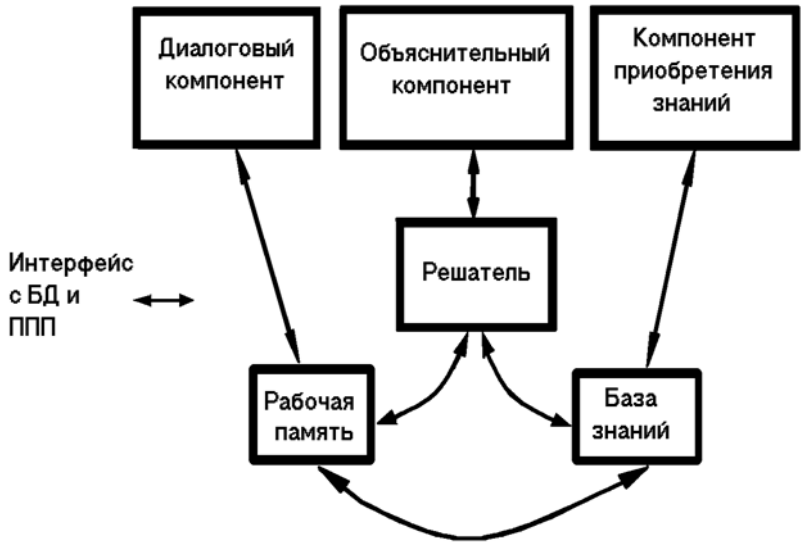


Рисунок6 –Схемастатическойэкспертнойсистемы

*Диалоговый компонент*ориентирован на организацию дружественного общения с пользователем как в ходе решения задач, так и в процессе приобретения знаний и объяснения результатов работы.

Экспертные системы и системы искусственного интеллекта отличаются от систем обработки данных тем, что в них в основном используются символьный (а не числовой) способ представления, символьный вывод.

6.1 Лабораторная работа № 6. Разработка экспертной системы

Цель работы: Изучение приемов по созданию экспертной системы для выбранной предметной области.

Задание: Для выбранной предметной области,сформировать базу знаний, соответствующую следующим требованиям:

– включить не менее 12 правил, из которых не менее 7 – сложные правила;

– для описания правил использовать не менее 8 переменных;

– число циклов просмотра правил для прямой цепочки рассуждений должно составлять не менее 3;

– для обратной цепочки рассуждений должны быть логически выведены не менее 4 переменных, прежде чем будет определена переменная вывода;

– парупоследовательныхправил.

Вариантыпредметныхобластей:

1. ЭС по выбору модели сотового телефона (компьютера, фото камеры и т. д.) с необходимыми характеристиками.
2. ЭС по подбору косметики с учетом типа кожи.
3. ЭС определения психологического типа человека.
4. ЭС определения продолжительности жизни человека в зависимости от образа жизни.
5. ЭС определениярастений (животных).
6. ЭС поиска неисправностей автомобиля и с выдачей рекомендаций по их устранению.
7. ЭС по кредитованию населения, предприятия.
8. ЭС по выбору временной формы английского глагола с учетом залога.

Вопросы для собеседования при защите работы:

1. Чтотакоеэкспертнаясистема?
2. Структура и назначение экспертных систем.
3. Основные области применения экспертных систем.
4. Перечислите основные типы экспертных систем.
5. Перечислите основные части экспертной системы.
6. Нарисуйте схему работы экспертной системы.
7. Перечислите основные этапы технологии разработки экспертных систем.
8. Перечислите основные разновидности систем искусственного интеллекта.

7 Математический нейрон

Мозг человека состоит из белого и серого вещества: белое – тела нейронов, серое – соединяющие их нервные волокна. Каждый нейрон состоит из трех частей: тела клетки, дендритов и аксона. Нейрон получает информацию через дендриты, а передает ее дальше через аксон, разветвляющийся в конце на тысячи синапсов – нервных нитей, соединяющих нейроны между собой. Каждый нейрон связан с 1000-10000 другими нейронами. Каждый нейрон может существовать в двух состояниях – возбужденном и невозбужденном. В возбужденное состояние нейрон переходит под действием электрических сигналов, поступающих от других нейронов, когда эти воздействия становятся достаточно большими.

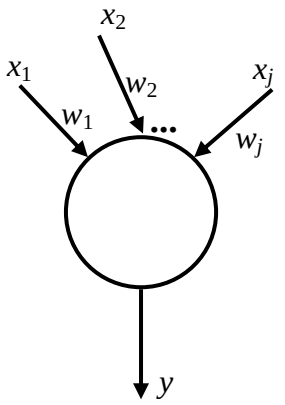


Рисунок 7 – Нейрон

Математическая модель искусственного нейрона была предложена Уорреном Мак-Каллоком и Вальтером Питтсом в 1943 г.вместе с моделью сети, состоящей из этих нейронов. Авторы показали, что сеть на таких элементах может выполнять числовые и логические операции. Математический нейрон имеет несколько входов и один выход (рисунок 7).

Через входы, число которых обозначим J, математический нейрон принимает входные сигналы x_j, которые суммирует, умножая каждый входной сигнал на некоторый весовой коэффициент w_j: $S = \sum_{j=1}^{J} w_j x_j$.

Выходной сигнал нейрона y может принимать одно из двух значений – 0 или 1, которые формируются следующим образом:

$$y = \begin{cases} 1, \text{если } S \geq \theta \\ 0, \text{если } S < \theta \end{cases}, (*)$$

где θ – порог чувствительности нейрона.

Логическая функция(*), называемая активационной функцией нейрона, ее график представлен на рисунке 8.

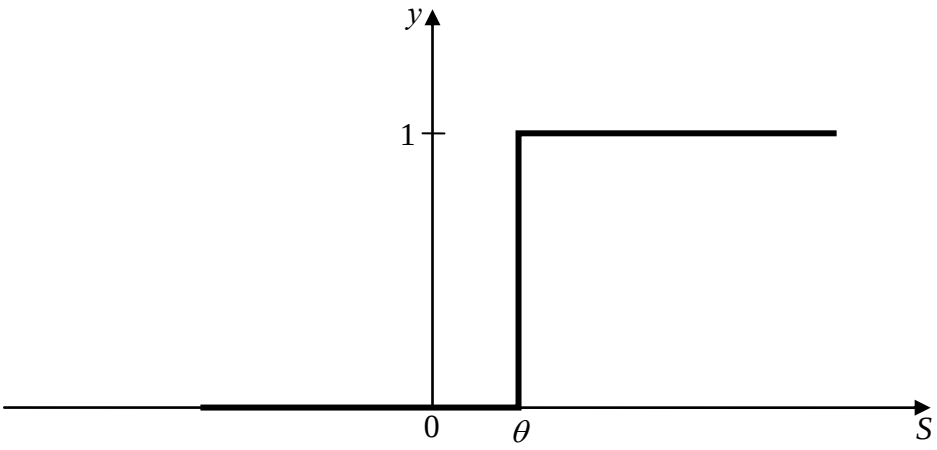


Рисунок 8 –Активационнаяфункциянейрона

Математические нейроны могут реализовывать различные логические функции. Так нейрон, имеющий два входа с единичными силами синаптических связей $w_1 = w_2 = 1$,реализует функцию логического умножения «И» при $\theta = 2$, и функцию логического сложения «ИЛИ» при $\theta = 1$. Нейрон с одним входом, у которого w=-1, реализует логическую функцию «НЕТ» при $\theta = 0$ (рисунок 9).

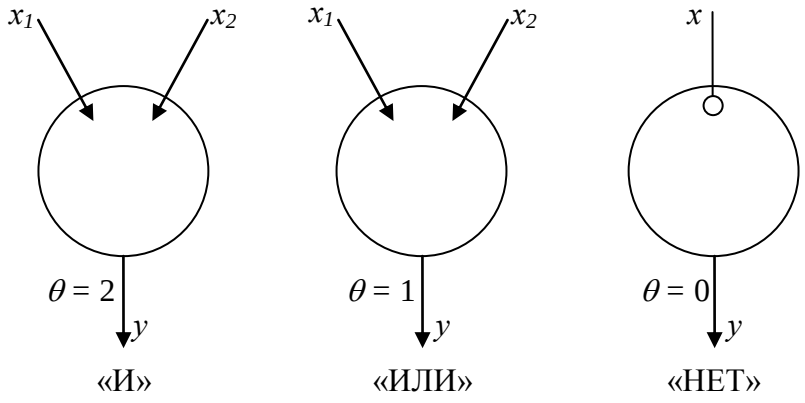


Рисунок9 – Нейроны, реализующиелогическиефункции

Иногда вместо понятия порога чувствительности нейрона θ используют термин нейронное смещенияb, которое отличается от порога θ только знаком:$b = -\theta$. Если величину bдобавить к сумме: $S = \sum_{j=1}^{J} w_j x_j + b$ то пороговая

активационная функция нейрона примет вид: $y = \begin{cases} 1, \text{если } S \geq 0 \\ 0, \text{если } S < 0 \end{cases}$.Графическое представление этой активационной функции приведено на рисунке 10:

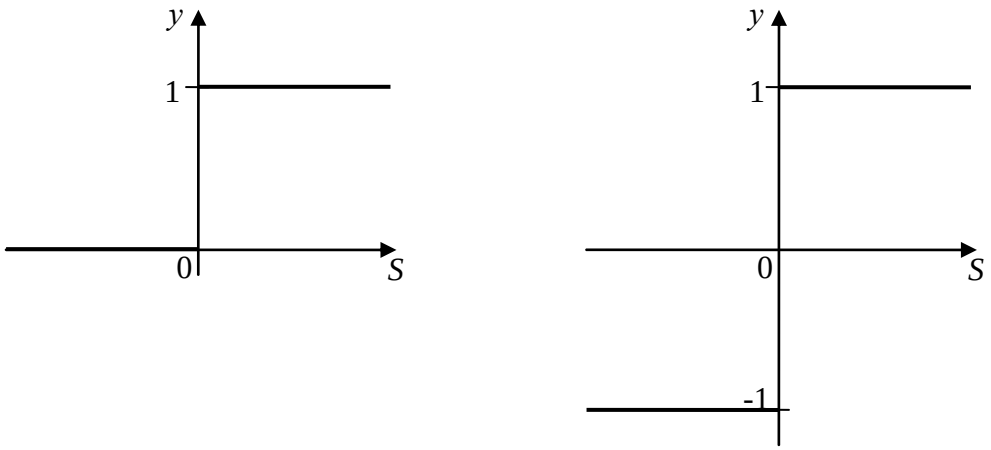


Рисунок 10 – Активационная функция нейрона с нейронным смещением

Нейронное смещение b можно рассматривать как весw_0 некоторого дополнительного входного сигнала x_0, величина которого всегда равна единице:

$$S = \sum_{j=1}^{J} w_j x_j + w_0 x_0 = \sum_{j=0}^{J} w_j x_j .$$

Нейрон с дополнительным входом x_0 изображен на рисунке 11.

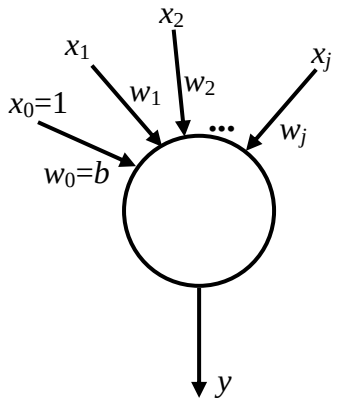


Рисунок 11 – Нейрон с дополнительным входом

7.1 Лабораторная работа № 7. Математический нейрон

Цель: Закрепление теоретических знаний и получение практических навыков работы с нейронами.

Задание: Разработать, используя язык C/C++, программу, реализующую работу нейрона Мак-Каллока – Питтса. Количество входов нейрона равно 3.

Вопросы для собеседования при защите работы:

1. Перечислите основные структурные элементы биологического нейрона.
2. Кем и когда была предложена математическая модель искусственного нейрона?
3. Что такое нейронное смещение?
4. Что такое активационная функция нейрона?
5. Как выглядит пороговая функция активации нейрона?

8 Однослойные нейронные сети

Формальный нейрон состоит из элементов 3 типов: умножителей (синапсов), сумматора и преобразователя (рисунок 12). Синапс характеризует силу (вес) связи между двумя нейронами. Сумматор выполняет сложение входных сигналов, предварительно помноженных на соответствующие веса. Преобразователь реализует функцию одного аргумента – выхода сумматора. Эта функция называется функцией активации или передаточной функцией нейрона. Исходя из данного описания, математическая модель нейрона может быть представлена следующим образом:$y = f(S),\ S = \sum_{j=1}^{J} w_j x_j + b.$

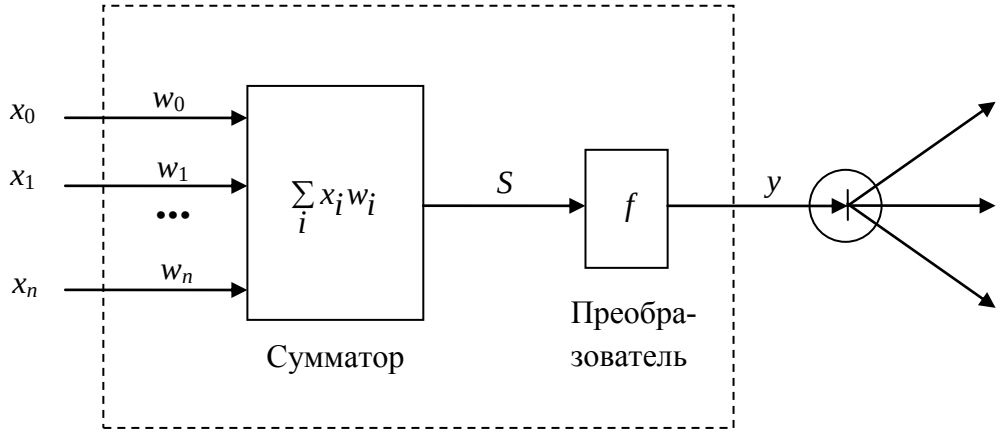


Рисунок12 –Структуранейрона

Примеры некоторых активационных функций представлены в таблице 1 и на рисунке13: а) пороговая; б) линейная; в) лог-сигмоидная; г) гиперболический тангенс.

Таблица 1 – Виды активационныхфункций

Название	Формула	Область значений
Пороговая	$f(S) = \begin{cases} 0, S < \Theta \\ 1, S \geq \Theta \end{cases}$	(0; 1)
Линейная	$f(S) = aS$	(-∞; +∞)

Продолжение таблицы 1

Название	Формула	Область значений
Лог-сигмоидная	$f(S)=\frac{1}{1+e^{-aS}}$	(0; 1)
Гиперболическийтангенс	$f(S)=\frac{e^{aS}-e^{-aS}}{e^{aS}+e^{-aS}}$	(-1; 1)

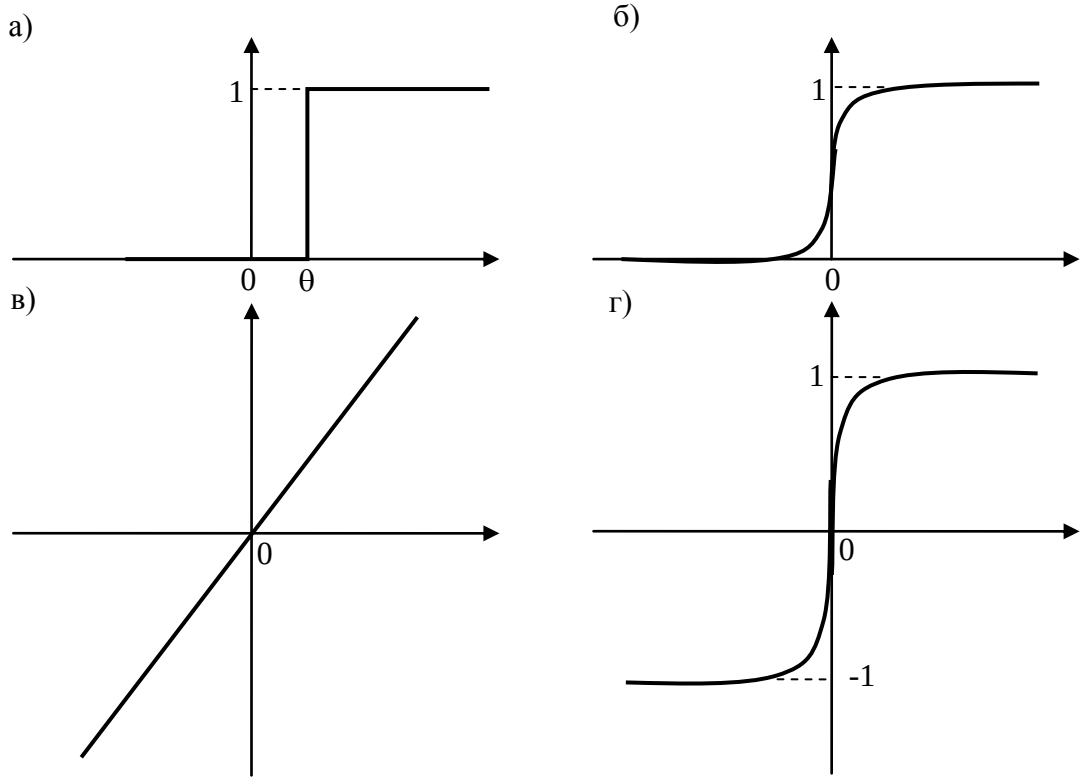


Рисунок13 –Видыактивационныхфункций

Формальные нейроны можно объединять таким образом, что выходные сигналы одних нейронов являются входными для других. Полученное множество связанных между собой нейронов называют *искусственными нейронными сетями*.

Простейшая модель нейронной сети – однослойный персептрон.

Однослойный персептрон (персептрон Розенблатта) – однослойная нейронная сеть, все нейроны которой имеют жесткую пороговую функцию активации (рисунок 14). Однослойный персептрон имеет простой алгоритм обучения и способен решать лишь самые простые задачи.

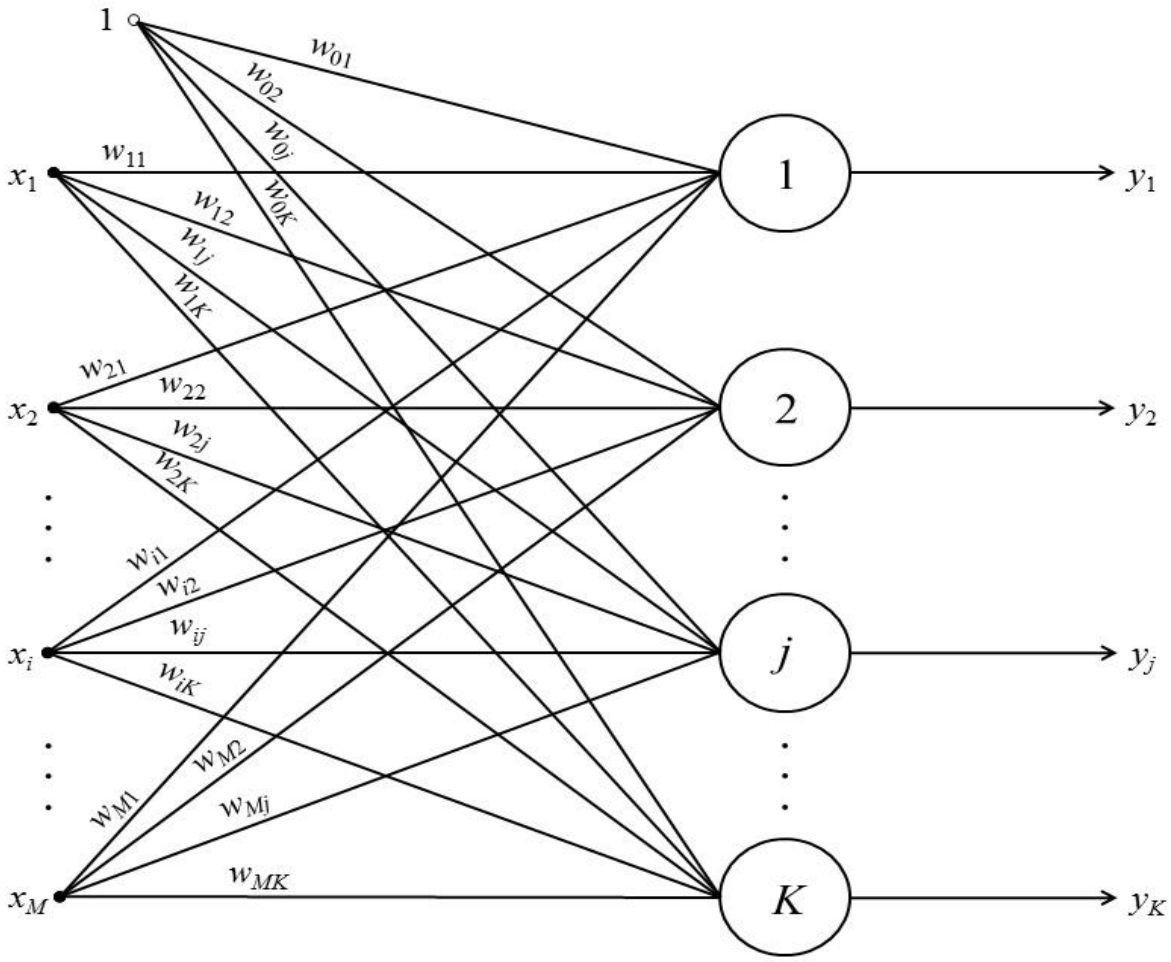


Рисунок 14 – Однослойный k-нейронный персептрон

Возможны несколько*типов классификации*существующих нейросетей. Наиболее часто используются следующие:

– *по типу входной информации*:сети, анализирующие двоичную информацию; сети, оперирующие с действительными числами;

– *по методу обучения*: сети, которые для того чтобы удовлетворять поставленным критериям требуют предварительного обучения перед включением их в реальную обстановку, – модели с учителем; сети, не требующие предварительного обучения, способные самообучаться (совершенствовать свои характеристики) в процессе работы;

– *по характеру распространения информации*: однонаправленные сети, в которых информация распространяется только в одном направлении от одного слоя

элементов к другому; рекуррентные сети, в которых выходной сигнал элемента может вновь поступать на этот элемент и другие элементы сети этого же или предыдущих слоев в качестве входного сигнала (сети с обратными связями);

– *по способу преобразования входной информации*: автоассоциативные; гетероассоциативные.

Обучение нейронной сети– поиск такого набора весовых коэффициентов, при котором входной сигнал после прохода по сети преобразуется в нужный нам выходной.

Шаг 1. Подать входной образ и вычислить выход персептрона y.

Шаг 2,а. Если выход правильный, то перейти на шаг 1.

Шаг 2,б. Если выход неправильный и = 0, то увеличить веса активных входов, например, добавить все входы к соответствующим им весам $w_j(t+1) = w_j(t) + x_j$.

Шаг 2,в. Если выход неправильный и равен 1, то уменьшить веса активных входов, например, вычесть каждый вход из соответствующего ему веса $w_j(t+1) = w_j(t) - x_j$.

Шаг 3. Перейти на шаг 1 или завершить процесс обучения.

В данном алгоритме шаг 2,б называют первым правилом Хебба, шаг 2,в – вторым правилом Хебба в честь ученого, предложившего этот алгоритм в 1949г.

Возникает вопрос: всегда ли алгоритм обучения персептрона приводит к желаемому результату? Ответ в следующей теореме:

Если существует множество значений весов, которые обеспечивают конкретное различение образов, то в конечном итоге алгоритм обучения персептрона приводит либо к этому множеству, либо к эквивалентному ему множеству, такому что данное различение образов будет достигнуто.

Дельта-правило

Дельта-правило является математической моделью правил корректировки весов. Введем величину ε, которая равна разности между требуемым d и реальным значением y, вычисляемым на его выходе: ε = (d – y). Тогда случай ε = 0 соответствует шагу 2,а, когда выход правилен, случай ε> 0 шагу 2,б, ε< 0 шагу 2,в.

Идея обучения персептрона с помощью правил Хебба сохранится, если итерационный процесс вести по формулам: $w_j(t+1) = w_j(t) + \Delta w_j,\ \Delta w_j = \varepsilon x_j$, где $w_j(t)$ и $w_j(t+1)$ соответственно старое и новое значение весовых коэффициентов, $j-$ номер входного сигнала. Кроме того, можно получить аналогичную итерационную формулу для подстройки порогового значения нейрона θ, если учесть, что его можно интерпретировать, как вес дополнительного входа x_0, значение которого равно -1: $\theta(t+1) = \theta(t) + \Delta\theta,\ \Delta\theta = -\varepsilon$. В итерационные формулы можно ввести коэффициент скорости обучения η, с помощью которого можно управлять величиной коррекции весов: $\Delta w_j = \eta\varepsilon x_j$, $\Delta\theta = -\eta\varepsilon$.

Наглядной иллюстрацией ограниченности для однослойного персептрона является функция "исключающее ИЛИ". Эта булева функция от двух переменных принимает значение "истина", когда значения входных переменных различны, и "ложь" – в противном случае. (Другими словами она принимает значение единицы, когда один из аргументов равен единице (но не оба).)Никакая комбинация значений двух весов не может дать соотношения между входом и выходом, задаваемого данной функцией. Это означает, что какие бы значения ни приписывались весам и порогу, сеть неспособна воспроизвести соотношение между входом и выходом, требуемое для представления функции "исключающее ИЛИ".

8.1 Лабораторная работа № 8. Обучение однослойной нейронной сети

Цель: Закрепление теоретических знаний и получение практических навыков работы с однослойныминейросетями.

Задание: Разработать, используя язык C/C++, программу, реализующую работу однослойнойнейросети. Количество нейронов равно 3. Обучить сеть логическим операциям «И», «ИЛИ».

Вопросы для собеседования при защите работы:

1. Перечислите основные структурные элементы математического нейрона.

2. Перечислите наиболее часто используемые активационные функции нейронов.

3. Какую область значений имеет логистическая функция активации нейрона?

4. Какую область значений имеет пороговая функция активации нейрона?

5. Перечислите основные типы нейронных сетей с точки зрения их топологии.

6. Чем различаются алгоритмы обучения с учителем и без учителя?

7. СформулируйтетеоремуКолмогорова.

8. Чтотакоеперсептрон?

9. Приведите алгоритм обучения однонейронного персептрона.

10. Сформулируйте дельта-правило для однослойного персептрона.

11. Можно ли с помощью однослойного персептрона реализовать логическую функцию исключающего ИЛИ (XOR)?

9 Многослойные нейронные сети

Многослойный персептрон – нейронная сеть прямого распространения сигнала (без обратных связей), в которой входной сигнал преобразуется в выходной, проходя последовательно через несколько слоев.

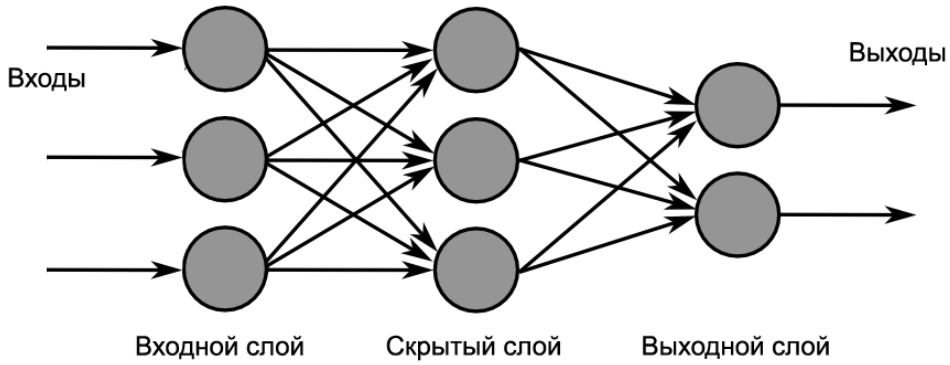


Рисунок 15 – Многослойная нейронная сеть с одним скрытым слоем

Первый из таких слоев называют входным, последний – выходным. Эти слои содержат так называемые вырожденные нейроны и иногда в количестве слоев не учитываются. Кроме входного и выходного слоев, в многослойном персептроне есть один или несколько промежуточных слоев, которые называют скрытыми.

В этой модели персептрона должен быть хотя бы один скрытый слой (рисунок 15). Присутствие нескольких таких слоев оправдано лишь в случае использования нелинейных функций активации (рисунок 16).

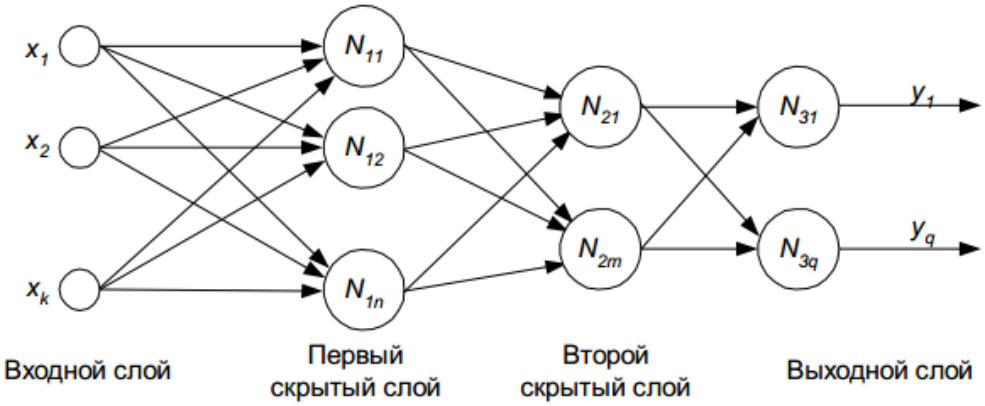


Рисунок 16 – Многослойная нейронная сеть с двумя скрытыми слоями

Обучение многослойнойнейросети

Алгоритм обратного распространения ошибки.

В настоящиймомент существует множество алгоритмов обучения. Самым распространенным является алгоритм обратного распространения ошибки, ставший известным в 1986 году, благодаря работе Румельхарта, Хилтона и Вильямса. Данный алгоритм опирается на обобщение дельта-правила, о котором мы говорили выше. Он используется для минимизации отклонения реальных значений выходных сигналов нейронной сети от требуемых. Функцию ошибки нейросетибудем вычислять по следующейформуле:

$$E(w) = \frac{1}{2}\sum_i E_i = \frac{1}{2}\sum_{i,k}(f_{i,k} - y_{i,k}^T)^2, \quad (1)$$

где $f_{i,k}$ – значение выходного сигнала k-го выходного нейрона сети при подаче на её входыi-го набора обучающих данных, $y_{i,k}^T$ – требуемое значение выходного сигнала k-го выходного нейрона дляi-го набора данных для обучения[9]. Суммирование ведется по всем нейронам выходного слоя и по всем наборам данных из обучающей выборки. Обучение нейронной сети направлено на минимизацию функции $E(w)$. Минимизация методом градиентного спуска обеспечивает подстройку весовых коэффициентов следующим образом:

$$\Delta w_{ij}^{(q)} = \eta \frac{\partial E}{\partial w_{ij}}, \quad (2)$$

$\Delta w_{ij}^{(q)}$ – величина изменения веса связи, соединяющейi-й нейрон (q–1) слоя с j-м нейроном слоя q; η – коэффициент скорости обучения, $0<\eta<1$. Таким образом, вес связи изменяется пропорционально ее вкладу в значение ошибки нейрона, для которого эта связь является входной, т.к. частная производная по весу $\frac{\partial E}{\partial w_{ij}}$ показывает зависимость скорости изменения функции ошибки E от изменения этого веса [9].

Изменение веса связи определяется следующим образом (3) (эта формула получается из формулы (2)):

$$\Delta w_{ij}^{(q)} = -\eta \delta_j x_i, \tag{3}$$

где δ_j – значение ошибки j-го нейрона в слое q, x_i – значениеi-го входного сигнала для j-го нейрона слоя q. Данную формулуможно использовать и для настройки смещений нейронов, только вместо x_i нужно подставить «1».

Значение ошибки нейрона определяется в зависимости от его положения в сети.

Для нейронов выходного слоя

$$\delta_i = (f_{i,k}(S))'(f_{i,k} - y_{i,k}), \tag{4}$$

где $y_{i,k}$– требуемое, а $f_{i,k}$ – фактическое значение выходного сигнала k-го нейрона для i-го набора данных из обучающей выборки, $(f_{i,k}(S))'$ значение производной активационной функции k-го нейрона для i-го набора обучающих данных [9].

Если нейрон принадлежит одному из скрытых слоев, то

$$\delta_i^{(q)} = (f_i^{(q)}(S))'\sum_j w_{ij}\delta_j^{(q+1)}, \tag{5}$$

где $\delta_i^{(q)}$ – ошибка i-го нейрона в слое q, $\delta_j^{(q+1)}$ – ошибкаj-го нейрона в (q+1) слое, w_{ij} – вес связи, соединяющей эти нейроны, $f_i^{(q)}$ – значение производной активационной функции i-го нейрона слоя q. Получаем, значение ошибки нейрона пропорционально его «влиянию» на величины ошибок нейронов следующего слоя, а также скорости изменения его выходного сигнала для k-го набора обучающих данных [9].

Рассмотрим нейросеть, состоящую из нейронов с лог-сигмоидными функциями активации:

$$f(S) = \frac{1}{1+e^{-aS}}, \tag{6}$$

где a – константа, S – взвешенная сумма входных сигналов нейрона, тогда

$$f'(S) = af(S)(1 - f(S)), \tag{7}$$

и формулы (4), (5) соответственно будут иметь вид

$$\delta_i = af_{i,k}(1 - f_{i,k})(f_{i,k} - y_{i,k}), \tag{8}$$

$$\delta_i^{(q)} = af_i(1 - f_i)\sum_j w_{ij}\delta_j^{(q+1)}. \tag{9}$$

Для реализации алгоритма обратного распространения ошибки может быть использована следующая последовательность действий:

1. Подать данные на вход нейронной сети.

2. Вычислить выходной сигнал сети.

3. Определить величину ошибок нейронов выходного слоя по формуле (4) или (8), и нейронов скрытых слоев по формулам (5) или (9).

4. Скорректировать веса.

5. Если в обучающей выборке есть неиспользованные в данной эпохе наборы данных, то перейти на шаг 1.

6. Рассчитать ошибку сети по формуле (1). Если ошибка меньше заданной, то закончить обучение, иначе, начать новую эпоху обучения и перейти на шаг 1.

Очень важно отметить, что алгоритм обратного распространения ошибки может быть использован только для нейронных сетей, содержащих нейроны с дифференцируемой функцией активации. Т.е. рассмотренный алгоритм не применим для настройки сетей, состоящих из нейронов с пороговыми функциями активации.

Рассмотрим пример работы и обучения нейронной сети.

Пример 1. Данный пример основан на примере из [9].

Дана сеть с 2 входами, двумя скрытыми слоями, в каждом из которых по 2 нейрона, и одним выходом. Функции активации нейронов – лог-сигмоидные (6), $a = 1$. Скорость обучения η примем равной 0,8. Необходимо реализовать с помощью нейронной сети логическую операцию «ИЛИ». Таблица истинности представлена в таблице2, где $x1$ и $x2$ – входные переменные, а y – значение результата.

Таблица 2 – Таблица истинности для операции «ИЛИ»

$x1$	$x2$	y
0	0	0
0	1	1
1	0	1
1	1	1

Предположим, что на некотором этапе обучения получена сеть, соответствующая изображенной на рисунке 17.

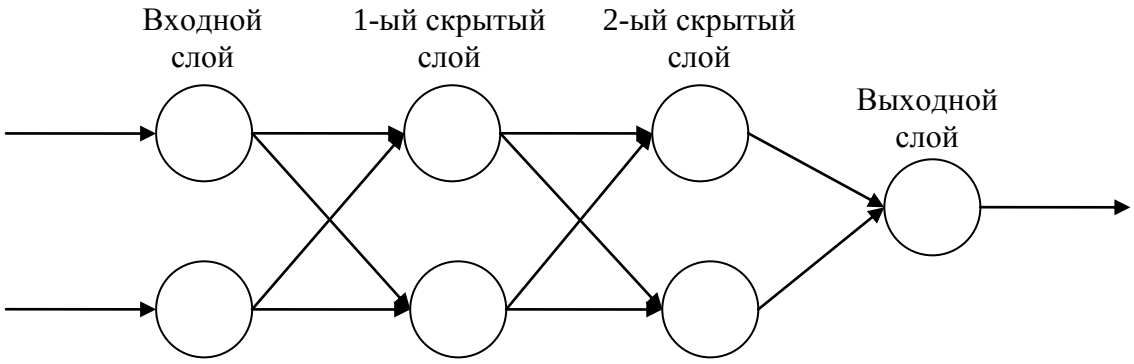


Рисунок 17 – Нейронная сеть, соответствующая примеру 1

Пусть на некотором шаге веса сети равны значениям из таблицы3. Смещения нейронов обозначены как веса с индексом «–».

Таблица 3 – Веса связей нейронной сети

Номер слоя	Индекс нейрона	Индекс веса	Вес
1	0	-	0,03
1	0	0	0,15
1	0	1	0,25
1	1	-	-0,01
1	1	0	-0,55
1	1	1	0,25
2	0	-	-0,08
2	0	0	-0,35
2	0	1	0,26
2	1	-	0,03
2	1	0	-0,07
2	1	1	0,75
3	0	-	0,05
3	0	0	0,06
3	0	1	0,65

Роль обучающей выборки в рассматриваемом случае играют данные из таблицы 2, в которой каждая строка соответствует одному набору данных. Пусть на вход нейронной сети поступили входные сигналы $x1 = 0$, $x2 = 1$. В соответствии с таблицей истинности (таблица2), выходной сигнал сети должен быть равен 1. Полученные значения выходов нейронов приведены в таблице4. На рисунке 18, наглядно показаны вычисления выходных сигналов нейронной сети.

Таблица 4 – Значения выходных сигналов нейронов

Номер слоя	Индекс нейрона	Значение выхода
1	0	0,57
1	1	0,56
2	0	0,467
2	1	0,601
3	0	0,615

Требуемое значение выхода сети «1», следовательно, ошибка сети на данном этапе равна:

$E_2 = 0{,}5 * (1 - 0{,}615)^2 = 0{,}074.$

Ошибка выходного нейрона:

$\delta_0^{(3)} = 0{,}615 * (1 - 0{,}615) * (0{,}615 - 1) = - 0{,}091.$

Ошибки нейронов 2 скрытого слоя:

$\delta_0^{(2)} = 0{,}467 * (1 - 0{,}467) * (- 0{,}091) * 0{,}06 = - 0{,}001,$

$\delta_1^{(2)} = 0{,}601 * (1 - 0{,}601) * (- 0{,}091) * 0{,}65 = - 0{,}014.$

Ошибки нейронов 1 скрытого слоя:

$\delta_0^{(1)} = 0{,}57 * (1 - 0{,}57) * ((- 0{,}001) * (- 0{,}35) + (- 0{,}014) * (- 0{,}07)) = 0{,}0003,$

$\delta_1^{(1)} = 0{,}56 * (1 - 0{,}56) * ((- 0{,}001) * 0{,}26 + (- 0{,}014) * 0{,}75) = - 0{,}003.$

Вес первой связи выходного нейрона изменится, согласно (3), на следующую величину:

$\Delta w_{00}^{(3)} = - 0{,}8\ \delta_0^{(3)} f_0^{(2)} = - 0{,}8 * (- 0{,}091) * 0{,}467 = 0{,}034.$

Таким образом, новое значение веса будет равно:

$w_{00}^{(3)}(t+1) = w_{00}^{(3)}(t) + \Delta w_{00}^{(3)} = 0{,}06 + 0{,}034 = 0{,}094.$

Аналогично производится коррекция остальных весов связей нейронной сети:

$w_{10}^{(3)}(t+1)= 0,65 – 0,8 * (– 0,091) * 0,601 = 0,694;$

$w_{-0}^{(3)}(t+1)= 0,05 – 0,8 * (– 0,091) * 1 = 0,123;$

$w_{00}^{(2)}(t+1)= -0,35 – 0,8 * (– 0,001) * 0,57 = – 0,35;$

$w_{10}^{(2)}(t+1)= 0,25 – 0,8 * (– 0,001) * 0,56 = 0,25;$

$w_{-0}^{(2)}(t+1)= – 0,08 – 0,8 * (– 0,001) * 1 = – 0,079;$

$w_{01}^{(2)}(t+1)= – 0,07 – 0,8 * (– 0,001) * 0,57 = – 0,07;$

$w_{11}^{(2)}(t+1)= 0,75 – 0,8 * (– 0,014) * 0,56 = 0,756;$

$w_{-1}^{(2)}(t+1)= 0,03 – 0,8 (– 0,014) * 1 = 0,041;$

$w_{00}^{(1)}(t+1)= 0,15 – 0,8 * 0,0003 * 0 = 0,15;$

$w_{10}^{(1)}(t+1)= 0,25 – 0,8 * 0,0003 * 1 = 0,25;$

$w_{-0}^{(1)}(t+1)= 0,03 – 0,8 * 0,0003 * 1 = 0,03;$

$w_{01}^{(1)}(t+1)= – 0,55 – 0,8 * (– 0,003) * 0 = – 0,55;$

$w_{11}^{(1)}(t+1)= 0,25 – 0,8 * (– 0,003) * 1 = 0,248;$

$w_{-1}^{(1)}(t+1)= – 0,01 – 0,8 * (– 0,003) * 1 = – 0,0076.$

В случаях, когда вместо индекса начального нейрона связи стоит символ «–», производится настройка смещения нейрона. Т.е. запись $w_{-1}^{(2)}(t+1)$означает новое значение смещения у нейрона с индексом 1 во втором скрытом слое. Использованные для расчета значения ошибок выходов нейронов показаны на рисунке 19.

Значения выходных сигналов ИНС при повторном подсчете выходов нейронов сети при использованном для коррекции весов наборе обучающих данных представлены в таблице 5 и на рисунке 20.

Таблица 5 – Значения выходных сигналов нейронов после корректировки весов связей нейросети

Номер слоя	Индекс нейрона	Значение выхода
1	0	0,555
1	1	0,564
2	0	0,467
2	1	0,606
3	0	0,643

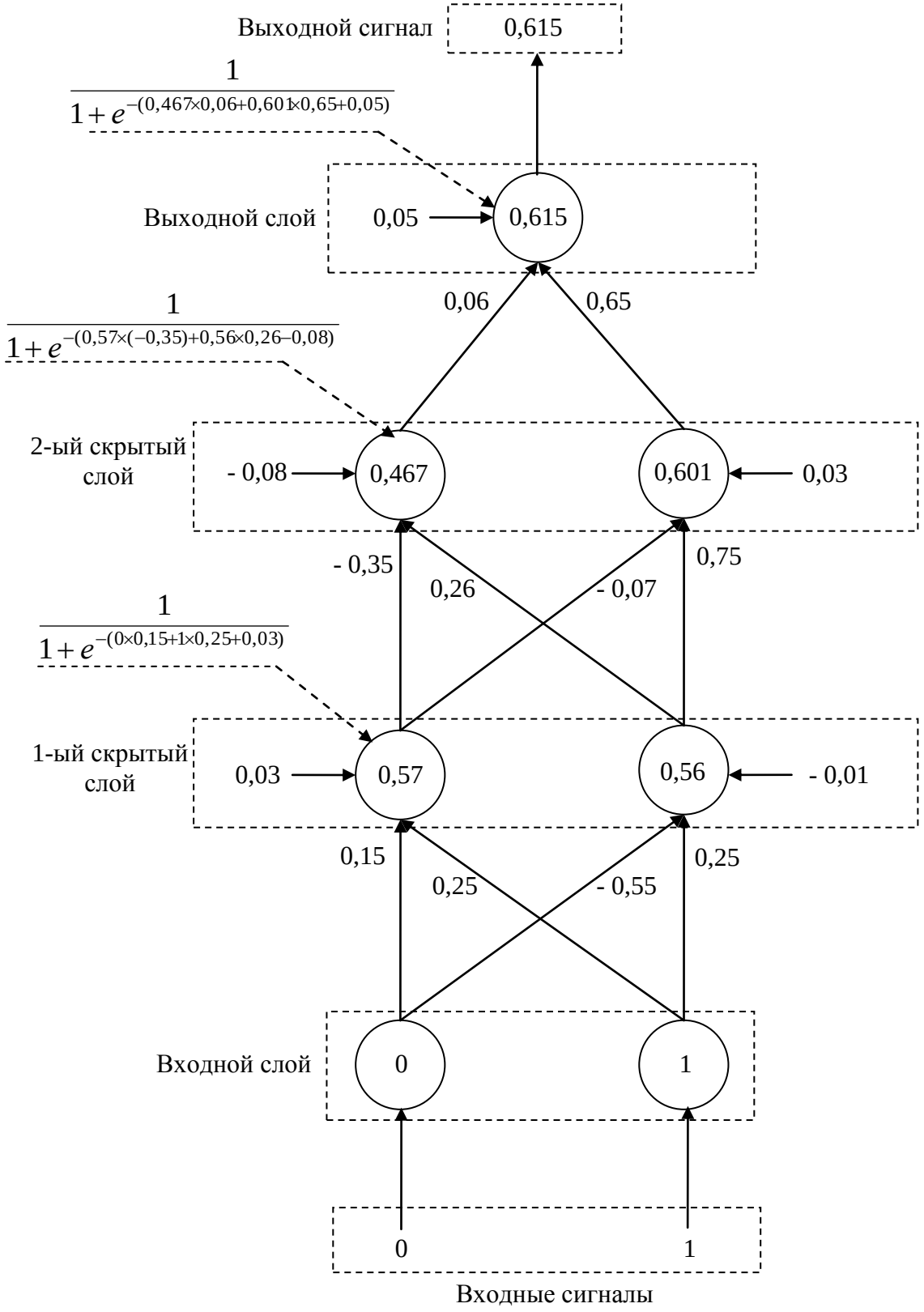


Рисунок 18 – Прямой проход для вычисления выхода нейросети

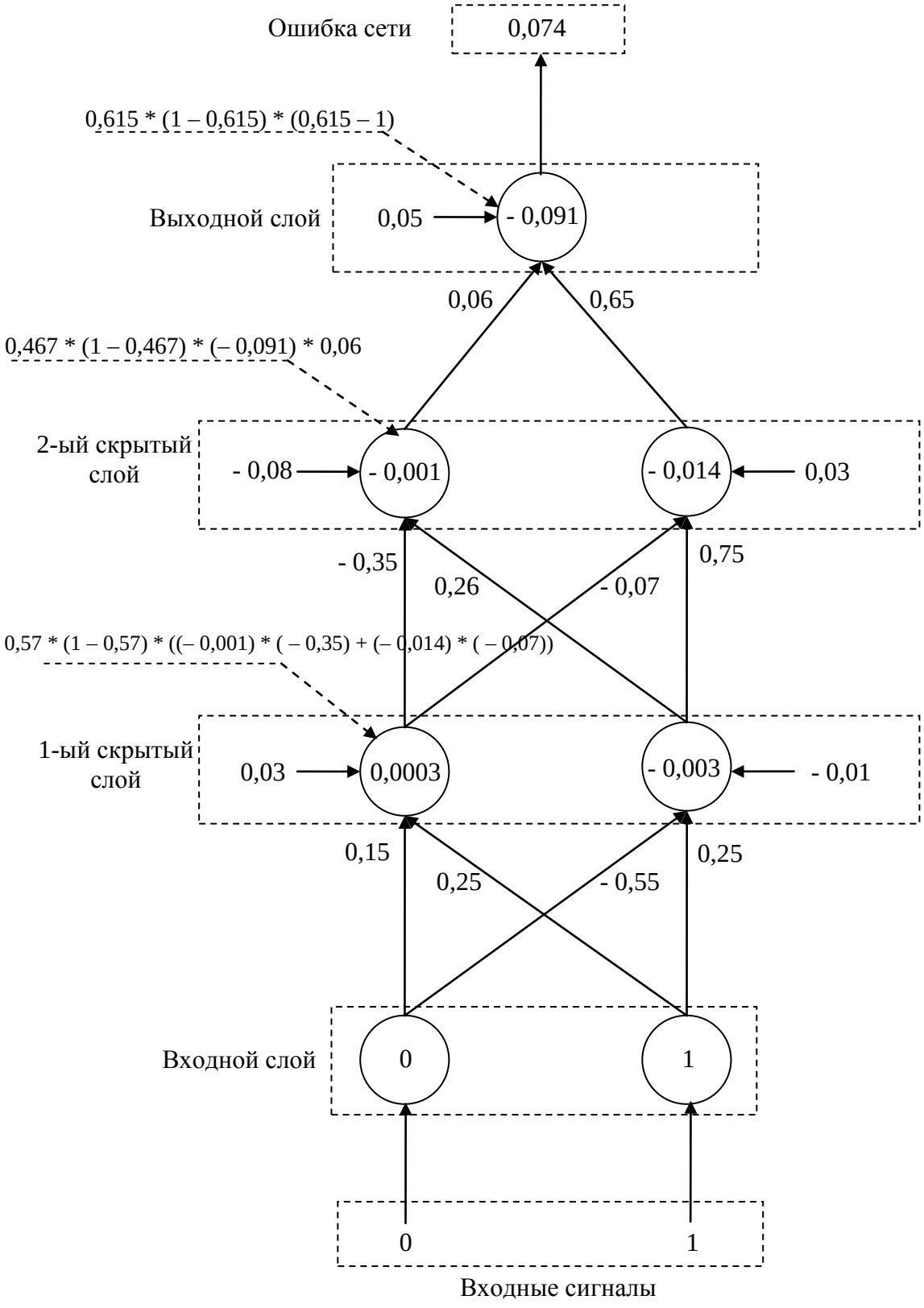


Рисунок 19 – Обратный проход для вычисления ошибок нейронов

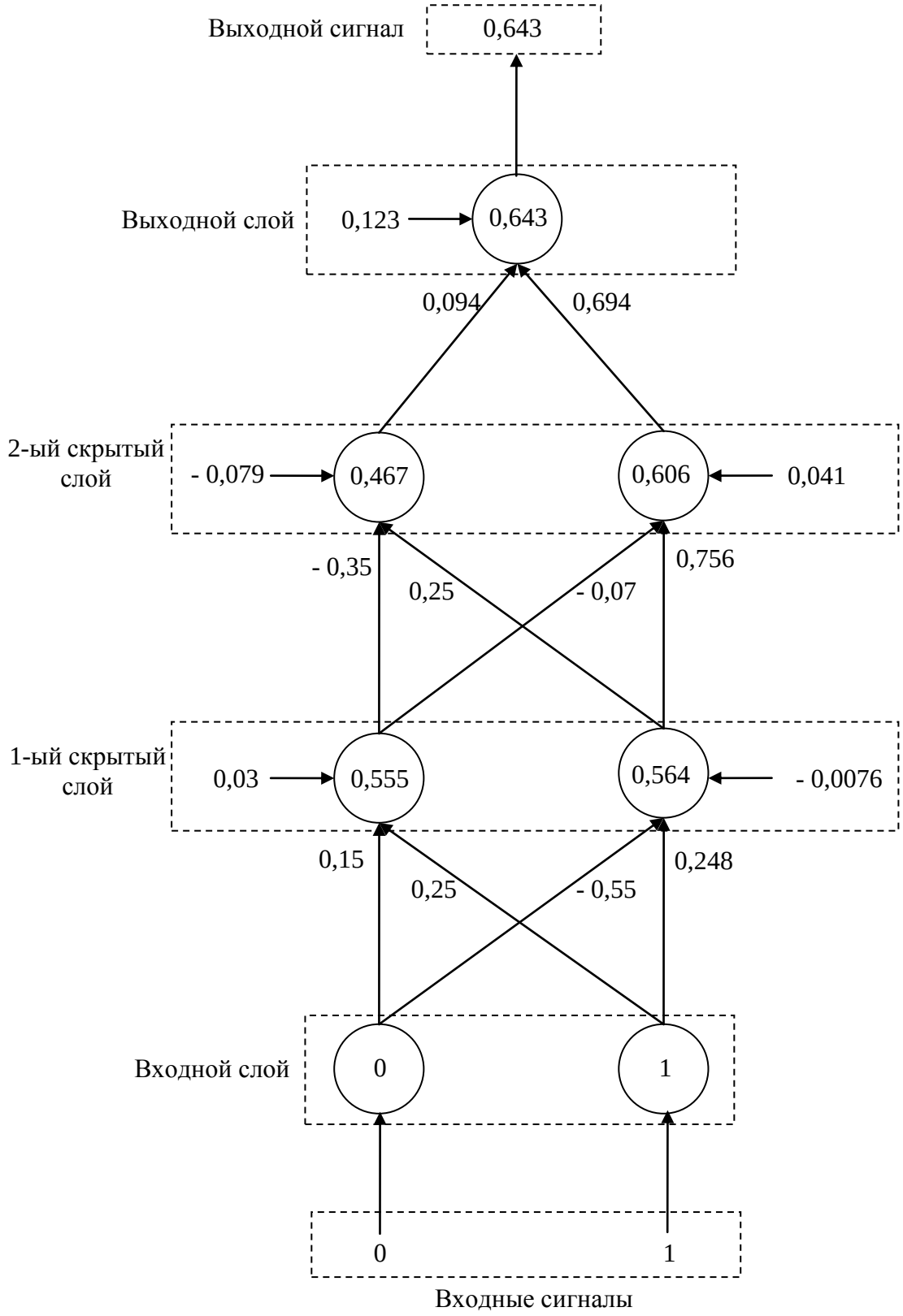


Рисунок 20 – Прямой проход после коррекции весов

Новое значение ошибки ИНС для 2-го набора обучающих данных равно:

$$E_2 = 0{,}5\ (1 - 0{,}643)^2 = 0{,}064.$$

Значение ошибки E_2 до коррекции весов было равно 0,074. Таким образом, в результате одного шага обучения значение ошибки уменьшилось на 0,01.

Процесс обучения нейросети продолжается. На вход подается следующий вектор данных из обучающего множества и производится коррекция весов связей с учетом расхождения выходного сигнала нейронной сети и вектора требуемых выходных сигналов, соответствующего входному. Обучение продолжается до тех пор, пока ошибка выходанейросети, вычисленная для всех наборов данных из обучающей выборки, по значению не будет меньше требуемой.

Рассмотрим вопросы, связанные с программной реализацией многослойной нейронной сети. При написании программы необходимо знать такиехарактеристики сети, как: число входов и выходов сети,число скрытых слоев, количество и тип нейронов в них [9]. Выше уже было сказано, что нейроны входного слоя в количестве слоев не учитываются, потому что, как правило, не выполняют никаких функций, а, значит, они не нуждаются в программной реализации. Будем считать, что входные сигналы сети совпадают с входными сигналами нейронов первого скрытого слоя, а если скрытых слоев нет, тосо входными сигналами нейронов выходного слоя. Необходимо заметить, что сети прямого распространения без обратных связей удовлетворяют следующим условиям: все нейроны одного слоя имеют одинаковые входные сигналы; входные сигналы слоя $(q+1)$ являются выходными сигналами слоя q (рисунок 21) [9].

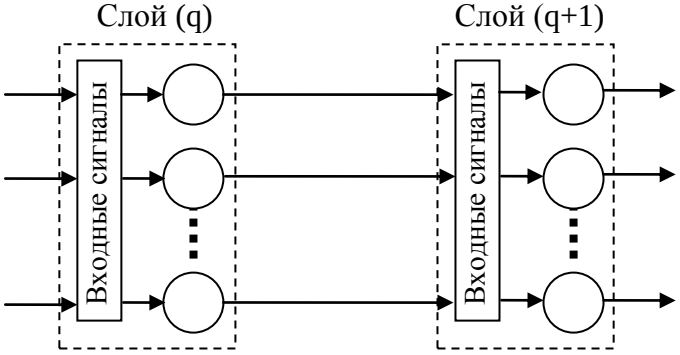


Рисунок 21 – Сети прямого распространения

Таким образом, каждый нейрон достаточно описать следующими параметрами: веса входящих связей (от нейронов предыдущего слоя к данному); смещение; коэффициент a в активационной функции; указатель на массив входных сигналов; указатель ссылка на элемент массива выходных сигналов слоя, которому принадлежит данный нейрон.

Слой нейроновв свою очередь можно считать нейронной сетью без скрытых слоев, поэтому важнейшими характеристиками для него будут:

- число входов – равно числу нейронов в предыдущем слое;
- число выходов – равно числу нейронов в данном слое.

В многослойной сети два смежных слоя (q) и (q + 1) имеют один общий массив, который будет содержать выходные сигналы слоя (q) и входные сигналы слоя (q+1). Поэтому целесообразно реализовать этот массив только один раз: либо как массив выходных сигналов слоя (q) (рис. 21), либо как массив входных сигналов слоя (q+1)[9].

В случае программной реализации искусственной нейронной сети с использованием объектно-ориентированного подхода можно выделить следующие объекты:

- нейрон;
- нейронный слой;
- нейронная сеть.

При этом класс каждого следующего объекта можно считать производным от класса предыдущего[9].

При структурном подходе нейронную сеть можно, например, представить как трехмерный массив, первый индекс которого обозначает слой, второй – нейрон в этом слое, а третий – вес соответствующей связи этого нейрона. Нейроны объединяются массивом, а не списком, так как это менее ресурсоёмко. Смещения нейронов могут быть заданы в этом же массиве, либо отдельно.

Например, сеть с двумя входами, одним выходом и двумя скрытыми слоями по два нейрона в каждом может быть представлена в виде массива следующим образом (на языке программирования C)[9]:

```
double ***net;            // массив весов связей сети
net = newdouble**[3];    // создание указателей на слои
net[0] = newdouble*[2];  // создание указателей  на нейроны 1-го слоя
net[1] = newdouble*[2];  // создание указателей  на нейроны 2-го слоя
net[2] = newdouble*[1];  // создание указателей   на нейроны 3-го слоя
net[0][0] = newdouble[2]; // создание массива весов  для 1-го нейрона 1-го слоя
net[0][1] = newdouble[2]; // создание массива весов для 2-го нейрона 1-го слоя
net[1][0] = newdouble[2]; // создание массива весов для 1-го нейрона 2-го слоя
net[1][0] = newdouble[2]; // создание массива весов  для 2-го нейрона 2-го слоя
net[2][0] = newdouble[2]; // создание массива весов  для 1-го нейрона 3-го слоя
```

Чтение/запись весов осуществляется обращением к массиву следующим образом:

```
doubletemp = net[0][1][1]; // чтение значения веса 2-го  сигнала 2-го нейрона 1-го слоя
net[1][0][1] = 0.12; // запись (изменение) значения веса 2-го сигнала 1-го нейрона 2-го слоя
```

Значения смещений можно задавать аналогичным образом через двумерный массив, в котором первый индекс обозначает слой, а второй – нейрон, к которому относится это смещение[9].

Важным понятием при обучении нейросети является понятие эпохи. При инициализации нейронной сети эта величина устанавливается равной 0 и имеет верхний предел, который можно задать вручную. Чем больше эпоха, тем лучше натренирована сеть и, соответственно, ее результат.

Прежде, чем программировать нейросеть, ее необходимо нарисовать для удобства. Тем самым определив количество слоев и количество нейронов в каждом слое.

Веса связей для нейронов можно задавать рандомно, сохраняя их в памяти для дальнейшей корректировки при обучении.

Необходимо определить какова будет функция активации нейронов в зависимости от решаемой задачи.

Также необходимо определить принцип остановки обучения нейросети. Это может быть определенное количество эпох (выше уже было сказано, что чем больше эпох, тем лучше), но лучше, если это будет конкретное значение ошибки, например, 0,001, дойдя до которого,нейросеть закончит процесс обучения.

9.1 Лабораторная работа № 9. Многослойные нейронные сети

Цель: Закрепление теоретических знаний и получение практических навыков работы с многослойныминейросетями.

Задание: Разработать, используя язык C/C++, программу, моделирующую поведение двухслойной нейронной сети и обеспечивающую ее обучение для решения задачи:

1) перекодирования прописных букв в строчные (маленькие – в большие). На вход сети подается код «маленькой» буквы, с выхода «снимается» код соответствующей «большой» буквы; (функция активации сигмоидная);

2) сжатия изображения (функция активации линейная).

Вопросы для собеседования при защите работы:

1. Чтотакоеадалайн?
2. Чтотакоемадалайн?
3. Что такое сеть обратного распространения?
4. Из каких слоев состоит сеть встречного распространения?
5. В чем заключается процедура нормализации?
6. Перечислите основные проблемы, решаемые с помощью искусственных нейронных сетей.

10 Распознавание образов

Распознавание образов (объектов, сигналов, ситуаций, явлений или процессов) – задача идентификации объекта или определения каких-либо его свойств по его изображению (оптическое распознавание) или аудиозаписи (акустическое распознавание) и другим характеристикам.

В системах искусственного интеллекта можно рассматривать два вида зрения:

Машинное зрение – направление исследований, ставящее своей целью моделирование зрительной информации восприятия, а в процессе функционирования накапливает образные и логические знания. Машинное зрение можно рассматривать и как имитацию и тренинг интеллектуальной деятельности человека.

Техническое зрение имеет узкоспециализированное назначение и содержит минимально необходимый набор датчиков информации и средств обработки для выполнения заданных функций.

Будем считать, что все объекты или явления разбиты на конечное число классов. Для каждого класса известно и изучено конечное число образов. Задача распознавания образов состоит в том, чтобы отнести новый распознаваемый объект к какому-либо классу.

Целью создания автоматизированных вычислительных систем распознавания является автоматизация группы процессов восприятия и познания, связанных с поиском, выделением, идентификацией, классификацией и описанием образов на основе анализа реальных данных.

Задачи распознавания – это информационные задачи, состоящие из двух этапов:

- преобразование исходных данных к виду, удобному для распознавания;
- собственно распознавание (указание принадлежности объекта определенному классу).

В целом проблема распознавания образов состоит из двух частей: обучения и распознавания.

Изучение и компьютерное моделирование процесса обучения является одним из предметов исследования в области искусственного интеллекта, называемой ***машинным обучением***.

В целом, можно выделить следующие методы распознавания образов:

- Метод перебора. В этом случае производится сравнение с базой данных, где для каждого вида объектов представлены всевозможные модификации отображения.
- Второй подход – производится более глубокий анализ характеристик образа (оптическое распознавание– определение различных геометрических характеристик,звуковое – частотный, амплитудный анализ и т. д.).
- Использование искусственных нейронных сетей (ИНС). Этот метод требует либо большого количества примеров задачи распознавания при обучении, либо специальной структуры нейронной сети, учитывающей специфику данной задачи. Но значительным преимуществом является более высокая эффективность и производительность.
- Экспертный метод, основанный на непрерывном обучении экспертной системы в процессе эксплуатации.

Рассмотрим основные понятия теории распознавания образов.

Воспринимаемые объекты ПО обычно разбиваются на группы похожих, но не тождественных объектов. Множество объектов, обладающих общими свойствами, называют ***образом***. Объекты, относящиеся к одному образу (классу), отличаются второстепенными, несущественными с точки зрения решаемой задачи, свойствами [5].

Каждый образ (класс) представляется некоторым количеством структурированных объектов – ***экземпляров*** [5].

Общее свойство объектов, объединенных в образ (класс), называется ***признаками образа*** (класса) или описанием и необходимо для установления соответствия образов, т.е. доказательства их идентичности, аналогичности, сходства, осуществляемого сопоставлением [5].

В классических моделях образ описывается вектором признаков, в структурной модели образ это высказывание, порождаемое грамматикой, характеризующей класс, которому данный образ принадлежит.

Образы формируются из простых стандартных элементов (абстрактные символы, множества, отношения или функции)[5].

Одним из базовых является понятие ***множества***. В компьютере множество представляется набором неповторяющихся однотипных элементов (какой-то элемент в множестве либо есть, либо его там нет). Универсальное множество включает все возможные для решаемой задачи элементы, пустое не содержит ни одного.

Методика отнесения элемента к какому-либо образу называется ***решающим правилом***.

Метрика– способ определения расстояния между элементами универсального множества. Чем меньше это расстояние, тем более похожими являются объекты (символы, звуки и др.) – то, что мы распознаем. Обычно элементы задаются в виде набора чисел, а метрика– в виде функции. От выбора представления образов и реализации метрики зависит эффективность программы, один алгоритм распознавания с разными метриками будет ошибаться с разной частотой.

Обучение– это процесс, в результате которого система постепенно приобретает способность отвечать нужными реакциями на определенные совокупности внешних воздействий, а ***адаптация***– это подстройка параметров и структуры системы с целью достижения требуемого качества управления в условиях непрерывных изменений внешних условий.

Во время ***распознавания*** (или ***классификации***) предлагаемые объекты распределяются по предопределенным классам.

Общая структура системы распознавания и этапы в процессе ее разработки показаны на рисунке22 [5].

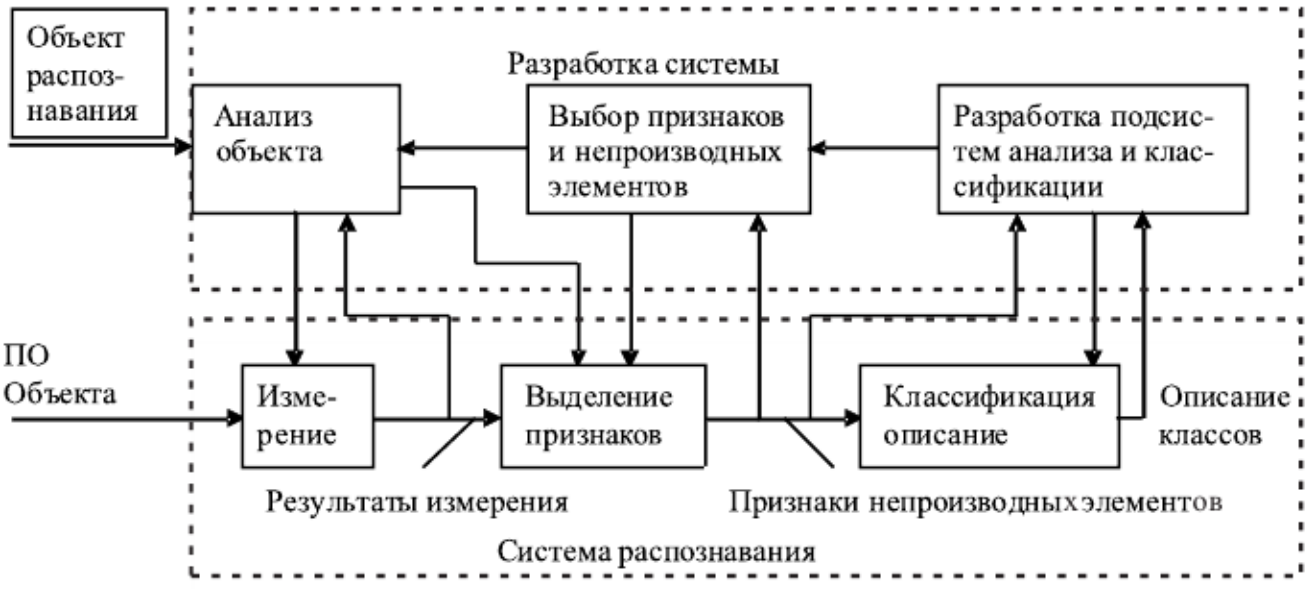


Рисунок 22 – Общая схема системы распознавания

Рассмотрим систему распознавания, использующую нейронные сети – персептрон Розенблатта.Он был первым электронным устройством, моделирующим человеческий глаз. Это устройство удалось обучить распознаванию букв латинского алфавита.

Термин «персептрон» был введен в 1950-х гг. Фрэнком Розенблаттом для некоторого класса интеллектуальных систем распознавания образов, способных обучаться на опыте. Первоначально персептрон Розенблатта (рисунок 23) содержал узлы трех типов. Сенсорные, или с-узлы, имитировали светочувствительные клетки

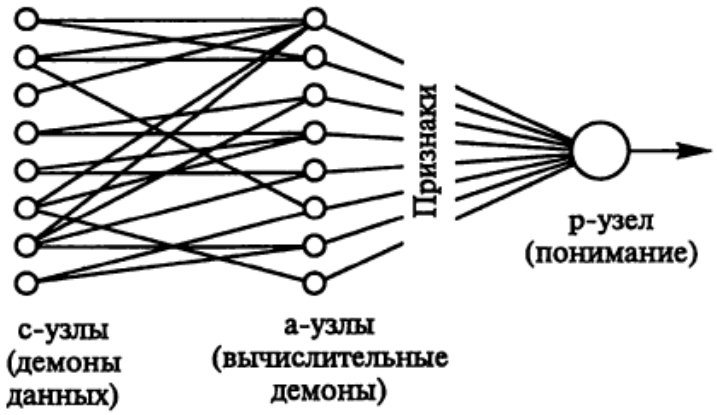


Рисунок 23 – Схема персептрона Розенблатта

сетчатки глаза. Они соответствовали демонам изображения, или демонам данных, пандемониумаСелфриджа. Обычно предполагается, что с-узлы являются элементами типа «все или ничего», но это не обязательно[11].

Следующий слой состоял из ассоциативных, или а-узлов, которые в общих чертах соответствуют вычислительным демонам пандемониума. В первоначальных вариантах исполнения персептрона соединения, идущие от с-узлов, формировались случайным образом еще в процессе конструирования системы, поэтому они определяли некоторые случайные свойства изображения. Как и в пандемониуме, при обучении персептрона вычислялись данные о ценности каждого а-узла. Входные соединения а-узла, ценность которых в процессе работы оказывалась малой, аннулировались, после чего случайным или псевдослучайным образом устанавливался новый набор соединений[11].

Выходы а-узлов были соединены с узлами реакции, или р-узлами, соответствующими демонам понимания пандемониума. В отличие от пандемониума р-узел дает только ответ «да» или «нет»[11].

Как а-узлы, так и р-узлы персептрона представляли собой математические нейроны. Некоторые из соединений между узлами являлись возбуждающими, а некоторые – тормозящими. Веса синапсов, идущих к р-узлам, изменялись в процессе обучения персептрона [11].

Алгоритм обучения персептрона состоял в следующем. Если реакция р-узла являлась правильной (т.е. он срабатывал, когда образ принадлежал к распознаваемому классу, или не срабатывал, когда образ не принадлежал указанному классу), то веса не изменялись.Если р-узел не срабатывал, когда распознаваемый образ на самом деле относился к рассматриваемому классу, то веса синапсов, бывших активными, увеличивались на некоторую величину с. С другой стороны, если р-узел срабатывал на образ, который не принадлежал распознаваемому классу, то веса активных синапсов уменьшались на величину с.С помощью этих весов вычислялись ценности каждого а-узла, и если ценность была мала, то его синапсы (связи с с-узлами) разрушались и строились новые.

Рассмотрим работу персептрона на примере классификации цифр на четные и нечетные [11].

Пример. Представим матрицу из 12 фотоэлементов, расположенных в виде 4 горизонтальных рядов, в каждом из которых 3 фотоэлемента. На матрицу фотоэлементов накладывается карточка с изображением цифры. Если на фотоэлемент попадает какой-либо фрагмент цифры, то данный фотоэлемент вырабатывает сигнал в виде двоичной единицы, в противном случае – 0.

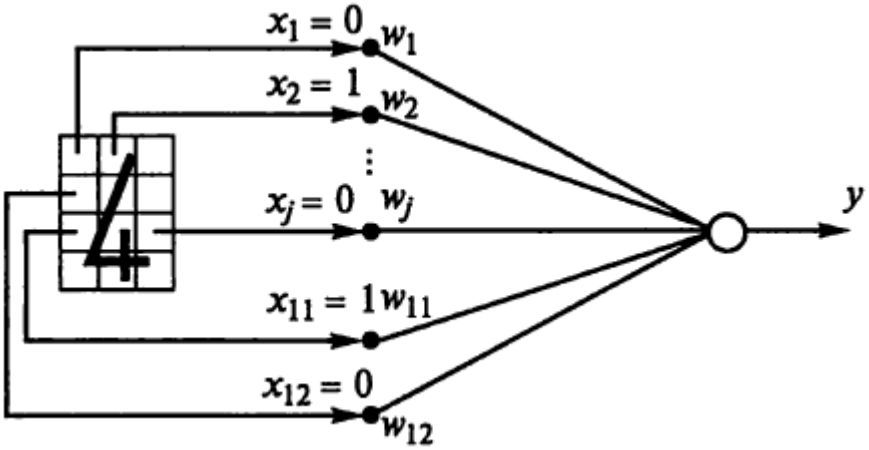


Рисунок 24 – Пример работы персептрона для классификации цифр на четные и нечетные

На рисунке 24 первый фотоэлемент выдает сигнал $x_1 = 0$, второй фотоэлемент $x_2 = 1$ и т.д. Согласно формуле $S = \sum_{j=1}^{J} w_j x_j$ персептронный нейрон выполняет суммирование входных сигналов x_j, помноженных на синаптические веса w_j, первоначально заданные датчиком случайных чисел. После этого сумма сравнивается с порогом чувствительности θ, так же заданным случайным образом.

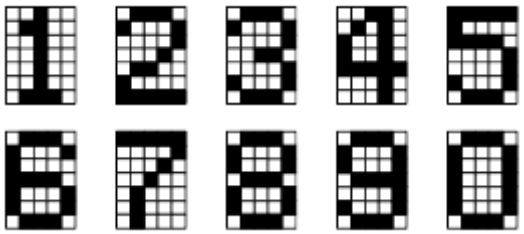
Цель обучения персептрона состоит в том, чтобы выходной сигнал был равен 1, если на карточке была четная цифра и 0, если цифра была нечетной. Эта цель достигается путем обучения персептрона, заключающемся в корректировке весовых коэффициетновw_j. Если была подана цифра 4 и выходной сигнал оказался равным 1, то корректировать ничего не нужно. Однако, если выход неправильный и $y = 0$, то

следует увеличить веса тех активных входов, которые способствуют возбуждению нейрона.

10.1 Лабораторная работа № 10. Распознавание образов с помощью персептронов

Цель: Закрепление теоретических знаний и получение практических навыков работы с многослойныминейросетями.

Задания: 1. Обучить нейросеть распознаванию цифр от 0 до 9, заданных в матричном виде 5x7 (рис.) (функция активации сигмоидная)



2. Обучить нейросеть распознаванию печатных латинских букв (функция активации сигмоидная).

3. Обучить нейросеть распознаванию геометрических фигур.

Вопросы для собеседования при защите работы:

1. Перечислите основные принципы, лежащие в основе распознающих систем.
2. Перечислите основные этапы распознавания образов.
3. Перечислите основные структурные элементы пандемониумаСелфриджа.
4. Какая связь между пандемониумомСелфриджа и персептроном?
5. Перечислите основные виды методов распознавания символов.
6. Перечислите известные вам системы распознавания текстов.
7. Геометрическийметодраспознавания.

11 Нечеткие множества и операции над ними

Нечеткое множество(fuzzyset) представляет собой совокупность элементов произвольной природы, относительно которых нельзя точно утверждать – обладают ли эти элементы некоторым характеристическим свойством, которое используется для задания нечеткого множества.

Пусть X – универсальное (базовое) множество, x – элемент X , а R – некоторое свойство. Обычное (четкое) подмножество A универсального множества X, элементы которого удовлетворяют свойству R, определяется как множество упорядоченных пар $A = \mu_A x/x$, где $\mu_A x$ – характеристическая функция, принимающая значение 1, если x удовлетворяет свойству R , и 0 – в противном случае.

Нечеткое подмножество отличается от обычного тем, что для элементов x из X нет однозначного ответа «да-нет» относительно свойства R. В связи с этим, нечеткое подмножество A универсального множества X определяется как множество упорядоченных пар $A = \mu_A x/x$, где $\mu_A x$ – *характеристическая функция принадлежности* (или просто *функция принадлежности*), принимающая значения в некотором вполне упорядоченном множестве $M = [0; 1]$. Функция принадлежности указывает степень (или уровень) принадлежности элемента x подмножеству A. Множество M называют множеством принадлежностей. Если $M = \{0; 1\}$, то нечеткое подмножество A может рассматриваться как обычное или четкое множество. Степень принадлежности $\mu_A x$ является субъективной мерой того, насколько элемент $x \in X$, соответствует понятию, смысл которого формализуется нечетким множеством A.

Носителем нечеткого множества A является четкое подмножество S_A универсального множества X со свойством $\mu_A x > 0$, т.е. $S_A = \{x | x \in X \wedge \mu_A x > 0\}$. Иными словами, носителем нечеткого множества A является подмножество S_A универсального множества X, для элементов которого функция принадлежности $\mu_A x > 0$ больше нуля. Иногда носитель нечеткого множества обозначают support$A$.

Величина $\sup_{x \in X} \mu_A x$ называется *высотой* нечеткого множества.

Нечеткое множество А*нормально*, если его высота равна 1, т.е. верхняя граница его функции принадлежности $\sup_{x\in X} \mu_A x = 1$. При $\sup_{x\in X} \mu_A x < 1$ нечеткое множество называется *субнормальным.*

Нечеткое множество называется *пустым*, если $\forall x \in X \mu_A x = 0$.

Непустое субнормальное множество всегда можно нормализовать, разделив все значения функции принадлежности на ее максимальное значение $\frac{\mu_A x}{\sup_{x\in X} \mu_A x}$.

Нечеткое множество называется *унимодальным*, если $\mu_A x = 1$ только для одной точки x (*моды*) универсального множества X .

Нечеткое множество называется *точечным*, если $\mu_A x > 0$ только для одной точки xуниверсального множества X .

Множеством α*-уровня* нечеткого множества A , определенного на универсальном множества X, называется четкое подмножество A_α универсального множества X, определяемое в виде:

$$A_\alpha = \{x \in X | \mu_A x \geq \alpha\} \text{ , где } \alpha \in [0 ; 1] .$$

Элементы $x \in X$, для которых $\mu_A x = 0{,}5$ называются *точками перехода* нечеткого множества A.

Ядром нечеткого множества A, определенного на универсальном множестве X, называется четкое множество *core*A, элементы которого удовлетворяют условию

$$core\ A = \{x \in X | \mu_A x = 1\}.$$

Границей нечеткого множества A, определенного на универсальном множестве X, называется четкое множество *front*A, элементы которого удовлетворяют условию

$$front\ A = \{x \in X | \ 0 < \mu_A x < 1\}.$$

Операции над нечеткими множествами

Включение. Пусть A и B – нечеткие множества на универсальном множестве X. Говорят, что A содержится в B, или B включает A, т.е. $A \subset B$, если $\forall x \in X \mu_A x \leq \mu_B x$. Иногда используют термин «доминирование», т.е. B доминирует A при $A \subset B$.

Равенство. Пусть A и B – нечеткие множества на универсальном множестве X. Говорят, что A и B равны, т.е. $A = B$, если $\forall x \in X \mu_A x = \mu_B x$. В противном случае $A \neq B$.

Дополнение. Пусть A и B – нечеткие множества с множеством принадлежностей характеристических функций $M = [0; 1]$, заданные на универсальном множестве X. Говорят, что A и B дополняют друг друга, т.е. $A = \overline{B}$ или $B = \overline{A}$, если $\forall x \in X \mu_A x = 1 - \mu_B x$. Очевидно следствие $\overline{\overline{A}} = A$ – так называемое свойство *инволюции*.

Пересечение нечетких множеств A и B, заданных на универсальном множестве X, – это наибольшее нечеткое множество $A \cap B$, содержащееся одновременно и в A, и в B с функцией принадлежности, заданной следующим образом:

$$\forall x \in X \mu_{A \cap B} x = min \{\mu_A x; \mu_B x\}.$$

Объединение нечетких множеств A и B, заданных на универсальном множестве X, – это наименьшее нечеткое множество $A \cup B$, включающее как A, так и B с функцией принадлежности, заданной следующим образом:

$$\forall x \in X \mu_{A \cup B} x = max \{\mu_A x; \mu_B x\}.$$

Разность нечетких множеств A и B , заданных на универсальном множестве X, – это нечеткое множество $A \setminus B = A \cap \overline{B}$ с функцией принадлежности, заданной следующим образом:

$$\forall x \in X \mu_{A \setminus B} x = \mu_{A \cap \overline{B}} x = min \{\mu_A x; 1 - \mu_B x\}.$$

Симметрическая разность нечетких множеств A и B, заданных на универсальном множестве X, – это нечеткое множество $A - B$ с функцией принадлежности, заданной следующим образом:

$$\forall x \in X \mu_{A\text{-}B} x = |\mu_A x - \mu_B x|.$$

Дизъюнктивная сумма нечетких множеств A и B, заданных на универсальном множестве X, – это нечеткое множество $A \oplus B = (A \setminus B) \cup (B \setminus A) = (A \cap \overline{B}) \cup (\overline{A} \cap B)$ с функцией принадлежности, заданной следующим образом:

$$\forall x \in X \mu_{A \oplus B} x = max\{min\{\mu_A x; 1 - \mu_B x\}; min\{1 - \mu_A x; \mu_B x\}\}.$$

11.1 Лабораторная работа № 11. Нечеткие множества и операции над ними

Цель: Изучение основных понятий, связанных с нечеткими множествами, и основных операций, производимых над нечеткими множествами.

Задание: Пусть на универсальном множестве X заданы следующие нечеткие подмножества:

$A = 0{,}2/x_1 + 0{,}4/x_2 + 1/x_3 + 0/x_4$,

$B = 0{,}7/x_1 + 0{,}9/x_2 + 1/x_3 + 0{,}1/x_4$,

$C = 0{,}1/x_1 + 0{,}1/x_2 + 0{,}2/x_3 + 0{,}9/x_4$.

1. Определить: ядро, границу, точки перехода для данных множеств, найти все подмножества α–*уровня*.
2. Найти: дополнение множеств*A*, *B*, *C*, пересечение, объединение, разность, симметрическую разность и дизъюнктивную сумму.

Вопросы для собеседования при защите работы:

1. Сформулируйтеопределениенечеткогомножества.
2. Что такое функция принадлежности нечеткого множества?
3. Сформулируйте принцип обобщения теории нечетких множеств.
4. Дайте развернутое определение операции объединения нечетких множеств.
5. Дайте развернутое определение операции пересечения нечетких множеств.
6. Что такое носитель нечеткого множества?
7. Когда нечеткое множество называется нормальным?
8. Что такое множество α-уровня для нечеткого множества?

12 Нечеткая логика

Определение. Любая нечеткая переменная характеризуется тройкой (x, U, X), где xназвание переменной, U– универсальное множество, X–нечеткое подмножество множества U, представляющее собой нечеткое ограничение на значение переменной $u \in U$, обусловленное x.

Лингвистическая переменная отличается от числовой переменной тем, что ее значениями являются не числа, а слова или предложения в естественном или формальном языке.

Определение. Лингвистическая переменная характеризуется набором свойств (X, $T(X)$, U, G, M), где:

X – название переменной;

$T(X)$обозначаеттерм-множествопеременнойX, т.е. множество названий лингвистических значений переменнойX, причем каждое из таких значений является нечеткой переменной $\tilde{x}$ со значениями из универсального множества U с базовой переменной u;

G –синтаксическое правило, порождающее названия $\tilde{x}$ значений переменной X;

M –семантическое правило, которое ставит в соответствие каждой нечеткой переменной $\tilde{x}$ ее смысл $M(\tilde{x})$, т.е. нечеткое подмножество $M(\tilde{x})$ универсального множества U.

Конкретное название $\tilde{x}$, порожденное синтаксическим правилом G, называется термом. Терм, который состоит из одного слова или из нескольких слов, всегда фигурирующих вместе друг с другом, называется атомарным термом. Терм, который состоит из более чем одного атомарного терма, называется составным термом.

Будем пользоваться термином «нечеткое высказывание» для обозначения утверждения вида «u естьA», где u название предмета, а A название нечеткого подмножества универсального множества U.

Значение истинности утверждения «*u* есть*A*», которое будем обозначать *v*(*A*) и определять как возможно нечеткое подмножество универсального множества значений истинности *V*. Будем предполагать, что *V*= [0,1].

Значение истинности, являющееся числом в [0,1], будем называть числовым значением истинности.

Значение истинности высказывания «*u* есть не*A*» определяется выражением $v(\overline{A}) = 1 - v(A)$.

Пусть *v*(*A*) и *v*(*B*) лингвистические значения истинности высказываний*A* и *B* соответственно. Согласно принципу обобщения, конъюнкция и дизъюнкция будут вычисляться по следующим формулам:

$$v(A) \wedge v(B) = v(A \text{ и } B) \Leftrightarrow \sup_{z=x \wedge y} (\mu_A(x) \wedge \mu_B(y))$$

$$v(A) \vee v(B) = v(A \text{ или } B) \Leftrightarrow \sup_{z=x \vee y} (\mu_A(x) \wedge \mu_B(y))$$

Операции конъюнкции и дизъюнкции, введенные Заде, определяются как:

$$x \wedge y = \min(x, y);$$
$$x \vee y = \max(x, y).$$

Пусть*A* и *B* нечеткие высказывания и μ_A, μ_B соответствующие им функции принадлежности. Тогда импликации $A \to B$ будет соответствовать некоторая функция принадлежности $\mu_{A \to B}(x, y) = \max(1 - \mu_A(x); \mu_B(y))$.

Нечетким логическим выводом (fuzzylogicinference) называется аппроксимация зависимости $Y = f(X_1, X_2, \ldots, X_n)$ каждой выходной лингвистической переменной от входных лингвистических переменных и получение заключения в виде нечеткого множества, соответствующего текущим значениям входов, с использованием нечеткой базы знаний и нечетких операций. Основу нечеткого логического вывода составляет композиционное правило Заде.

В общем случае нечеткий вывод решения происходит за три (или четыре) шага:

1) этап фаззификации (приведение к нечеткости);

2) этап непосредственного нечеткого вывода;

3) этап композиции (агрегации, аккумуляции).

4) этап дефаззификации (необязательный) (приведение к четкости). Мы будем использовать метод «по максимуму», гдев качестве точного значения выводимой переменной принимается максимальное значение функции принадлежности.

Рассмотрим пример использования аппарата нечеткой логики из [12].

Пример. Пусть у нас есть некоторая система, например, реактор, описываемая тремя параметрами: температура, давление и расход рабочего вещества. Все показатели измеримы, и множество возможных значений известно. Также из опыта работы с системой известны некоторые правила, связывающие значения этих параметров. Предположим, что сломался датчик, измеряющий значение одного из параметров системы, но знать его показания необходимо хотя бы приблизительно. Тогда встает задача об отыскании этого неизвестного значения (пусть это будет давление) при известных показателях двух других параметров (температуры и расхода) и связи этих величин в виде следующих правил:

- если Температура низкая и Расход малый, то Давление низкое;
- если Температура средняя, то Давление среднее;
- если Температура высокая или Расход большой, то Давление высокое.

В нашем случае Температура, Давление и Расход лингвистические переменные. Опишем каждую из них.

Температура. Универсум (множество возможных значений) отрезок [0,150]. Начальное множество термов {Высокая, Средняя., Низкая}. Функции принадлежности термов изображены на рисунке 25.

Давление. Универсум отрезок [0,100]. Начальное множество термов {Высокое, Среднее, Низкое}. Функции принадлежности изображены на рисунке 26.

Расход. Универсум отрезок [0, 8]. Начальное множество термов {Большой, Средний, Малый}. Функции принадлежности термов изображены на рисунке 27.

Рисунок 25 – Графики функций принадлежности термов лингвистической переменной «Температура»

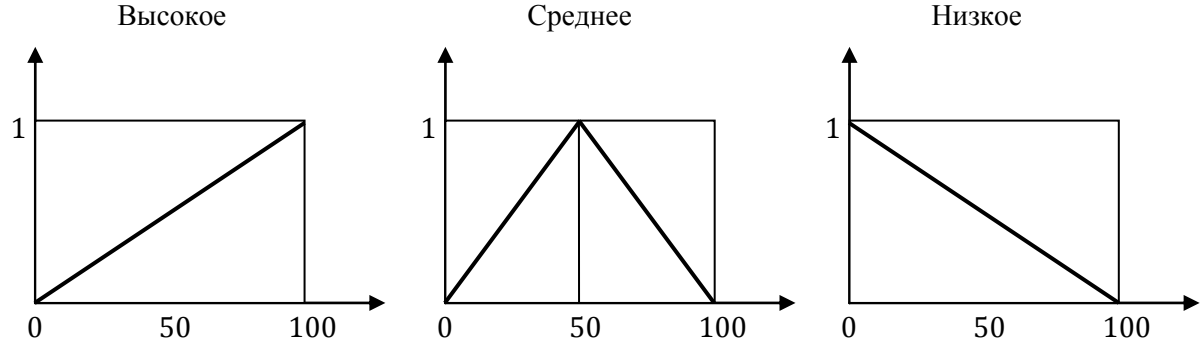


Рисунок 26 – Графики функций принадлежности термов лингвистической переменной «Давление»

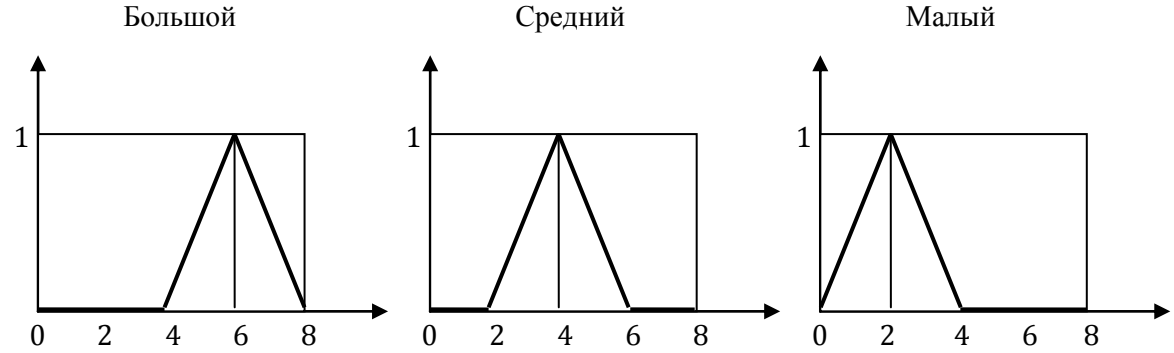


Рисунок 27 – Графики функций принадлежности термов лингвистической переменной «Расход»

Пусть известны значения Температура 85 и Расход 3,5 . Произведем расчет значения давления.

Последовательно рассмотрим этапы нечеткого вывода:

Фаззификация: Вычислим степень уверенности каждого терма:

Температура Высокая 0,7;

Температура Средняя 1;

Температура Низкая 0,3;

Расход Большой 0;

Расход Средний 0,75;

Расход Малый 0,25.

Вычислим степени уверенности посылок правил:

Температура низкая и Расход малый: min (Темп. Низкая, Расход Малый) = min(0.3, 0.25) = 0.25;

Температура Средняя: 1;

Температура Высокая или Расход Большой: max (Темп. Высокая, Расход Большой) = max(0.7, 0) = 0,7.

Нечеткий вывод: используя определение нечеткой импликации как минимума левой и правой частей (определение Mamdani), имеем (рисунок 28):

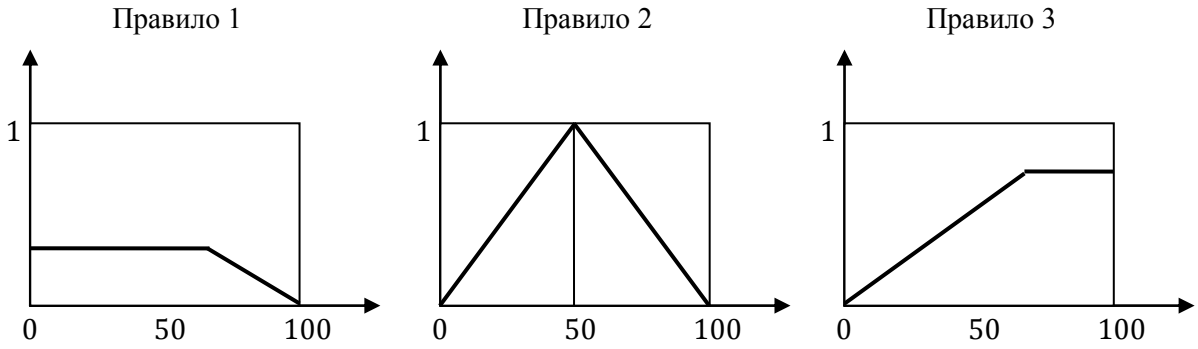


Рисунок 28 – Результат нечеткого логического вывода

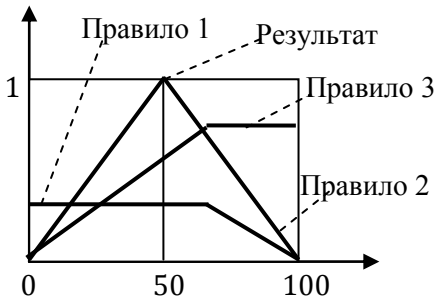


Рисунок 29 – Аккумуляция

Аккумуляция изображена на рисунке 29.

Таким образом, если мы знаем, что температура равна 85, а расход рабочего вещества 3,5, то можем сделать вывод, что давление в реакторе равно примерно 50.

12.1 Лабораторная работа № 12. Нечеткая логика

Цель: Изучение основных понятий, связанных с нечеткими множествами, и основных операций, производимых над нечеткими множествами.

Задание: 1. Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных), проверить ее на полноту и произвести нечеткий вывод для конкретных значений для задачи покупки фруктов (соотношение цены, качества, объёма покупки и т.д.);

2. Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных), проверить ее на полноту и произвести нечеткий вывод для конкретных значений для задачи управления движениями робота (направление движения, положение препятствия, обход препятствия);

3. Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных), проверить ее на полноту и произвести нечеткий вывод для конкретных значений для задачи управления наполнением воды в бассейне (начальное количество воды, погодные условия, приток воды).

Вопросы для собеседования при защите работы:

1. Приведите пример нечеткой операции отрицания.
2. Приведите пример нечеткой операции конъюнкции.
3. Приведите пример нечеткой операции дизъюнкции.
4. Примерызаписинечеткогомножества.
5. Графическоепредставлениенечеткихмножеств.
6. Нечеткийлогическийвывод.

Примерный вариант теста для рубежного контроля № 1

1. Искусственныйинтеллект – это:

а) область информатики, которая занимается разработкой интеллектуальных компьютерных систем, то есть систем, обладающих возможностями, которые мы традиционно связываем с человеческим разумом**;**

б) область информатики, которая занимается разработкой компьютерных систем, обладающих возможностями решать сложные математические задачи;

в) область информатики, которая занимается разработкой компьютерных систем, обладающих возможностями решать статистические задачи на разных языках программирования;

г) область информатики, которая занимается разработкой компьютерных систем, обладающих возможностями решать математические задачи на разных языках программирования.

2. Какой из языков программирования является языком логического программирования?

а) Prolog;

б) C#;

в) C++;

г) Lisp.

3. SS-проблемасостоит в:

а) поиске декомпозиции исходной задачи на подзадачи, приводящей к задачам, решение которых системе известно;

б) поиске пути, ведущего из начального состояния в одно из конечных;

в) поискенестандартногорешениязадачи;

г) поиске решения неразрешимых задач математики.

4. PR-проблемасостоит в:

а) поиске декомпозиции исходной задачи на подзадачи, приводящей к задачам, решение которых системе известно;

б) поиске пути, ведущего из начального состояния в одно из конечных;

в) поискенестандартногорешениязадачи;

г) поиске решения неразрешимых задач математики.

5. В рамках направления "Представление знаний" решаются задачи, связанные с:

а) реализацией различных подходов к их решению;

б) формализацией и представлением знаний в памяти человека-эксперта;

в) формализацииеймышлениячеловека;

г) формализацией и представлением знаний в памяти интеллектуальной системы.

6. Какие из моделей являются моделями представления знаний?

а) продукционныемодели;

б) семантическиесети;

в) фреймы;

г) логическиемодели;

д) математическиемодели.

7. Продукционные модели представления знаний задаются формулой:

а) $M = <T, P, A, B>$;

б) (i); Q; P; $A => B$; N;

в) $H = <I, C_1, C_2, ..., C_n, \Gamma>$;

г) $C_1, C_2, ..., C_n$.

8. Сетевые модели представления знаний задаются формулой:

а) $M = <T, P, A, B>$;

б) (i); Q; P; $A => B$; N;

в) $H = <I, C_1, C_2, ..., C_n, \Gamma>$;

г) $C_1, C_2, ..., C_n$.

9. Логические модели представления знаний задаются формулой:

а) $M = <T, P, A, B>$;

б) (i); Q; P; $A => B$; N;

в) $H = <I, C_1, C_2, ..., C_n, \Gamma>$;

г) $C_1, C_2, ..., C_n$.

10. Базазнаний – это:

а) формализованные данные о предметной области;

б) формализованные знания о предметной области и о том, как решать задачу;

в) база данных о предметной области;

г) совокупность всех знаний интеллектуальной системы.

11. Особенностямизнанийявляются:

а) внутренняяинтерпретируемость;

б) структурированность;

в) компактность;

г) пассивность.

12. Знания о предметной области, объектах этой области, их отношениях, действиях над ними называют:

а) предметныезнания;

б) прагматическиезнания;

в) семантическиезнания;

г) синтаксическиезнания.

13. Экспертнаясистема – это:

а) система, способная решать различные прикладные задачи;

б) система, машинногоперевода;

в) система, основаннаяназнаниях;

г) система, способная понимать данные на естественном языке.

14. Функцииэкспертныхсистем:

а) приобретениезнаний;

б) представлениезнаний;

в) решениестатистическихзадач;

г) решениематематическихзадач.

15. Составляющимиэкспертнойсистемыявляются:

а) базазнаний;

б) механизмвывода;

в) эксперт;

г) рабочаяпамять.

16. Классификация экспертных систем по динамичности делит ЭС на:

а) прямые и косвенные;

б) статические и динамические;

в) статические, динамические и сверхдинамические;

г) квазидинамические и динамические.

17. Отличие динамических ЭС от статических ЭС состоит в том, что в них:

а) учитываются изменения окружающего мира, происходящие за время решения задачи;

б) не учитываются изменения окружающего мира, происходящие за время решения задачи;

в) учитываются изменения, происходящие внутри самой ЭС;

г) не учитываются изменения, происходящие внутри самой ЭС

18. С учетом архитектуры экспертной системы знания целесообразно делить на:

а) простые и сложные;

б) интерпретируемые и неинтерпретируемые;

в) основные и вспомогательные;

г) активные и пассивные.

19. В каком году была создана первая экспертная система?

а) 1965;

б) 1990;

в) 2000;

г) 1997.

20. Какие компоненты отличают архитектуру динамической ЭС от статической?

а) подсистемамоделированиявнешнегомира;

б) базазнаний;

в) подсистема связи с внешним окружением;

г) решатель.

Примерный вариант теста для рубежного контроля № 2

1. Какие выделяют подходы к созданию искусственных нейронных сетей?

а) аппаратный и программный;

б) статический и динамический;

в) простой и составной;

г) аналитический и практический.

2. Какие этапы построения нейронных сетей выделяют?

а) выбортипанейроннойсети;

б) обучениенейроннойсети;

в) структуризациянейроннойсети;

г) оценкапараметровнейроннойсети.

3. Точка соединения дендрита и аксона называется:

а) многослойныйперсептрон;

б) адалайн;

в) мадалайн;

г) синапс.

4. Математическийнейрон – это:

а) нейроны мозга человека с математическим складом ума;

б) математическоеописаниемыслейчеловека;

в) нейрон, подчиняющийсязаконаммматематики;

г) математическая модель биологического нейрона мозга человека.

5. Изобретателемперсептронаявляется:

а) В. Питтс;

б) У. Мак-Каллок;

в) Ф. Розенблат;

г) М. Хофф.

6. Адалайн – это:

а) персептроны с сигмоидными активационными функциями с одним выходом;

б) персептроны с сигмоидными активационными функциями с несколькими выходами;

в) персептроны с экспоненциальными активационными функциями с одним выходом;

г) персептроны с экспоненциальными активационными функциями с несколькими выходами.

7. Мадалайн – это:

а) персептроны с сигмоидными активационными функциями с одним выходом;

б) персептроны с сигмоидными активационными функциями с несколькими выходами;

в) персептроны с экспоненциальными активационными функциями с одним выходом;

г) персептроны с экспоненциальными активационными функциями с несколькими выходами.

8. Под нейронной сетью обычно понимается:

а) структура из связанных между собой нейронов;

б) нервнаясистемачеловека;

в) системарешенияинтеллектуальныхзадач;

г) нетверногоответа.

9. Первое электронное устройство, моделирующее человеческий глаз называлось:

а) аксон;

б) дендрит;

в) пандемониум;

г) перцептрон.

10. Какие бывают типы (по организации обучения) нейронных сетей?

а) с учителем;

б) без учителя;

в) замкнутые;

г) открытые.

11. По характеру распространения информации нейронные сети делятся на:

а) однонаправленные и рекуррентные;

б) однослойные и многослойные;

в) замкнутые и открытые;

г) нет верного ответа.

12. Множество Первым алгоритмом обучения нейронной сети был:

а) Дельта-правило;

б) Обобщенное дельта-правило;

в) Адалайн;

г) Мадалайн.

13. В Какие правила легли в основу алгоритма обучения по дельта-правилу:

а) правила Хебба;

б) правила Розенблатта;

в) правила Уидроу;

г) нет верного ответа.

14. К областям применения нейронных сетей относят:

а) диагностика;

б) решение алгоритмических задач;

в) прогнозирование;

г) нет верного ответа.

15. Однослойная нейронная сеть, все нейроны которой имеют жесткую пороговую функцию активации называется:

а) адалайн;

б) мадалайн;

в) однослойный синапс;

г) однослойныйперсептрон.

16. Задача идентификации объекта или определения каких-либо его свойств по его изображению или аудиозаписи и другим характеристикам называется:

а) распознаванием образов;

б) классификацией объектов;

в) кластеризацией;

г) нет верного ответа.

17. Задача распознавания образов состоит в том, чтобы:

а) разбить множество объектов на классы;

б) построить компьютерную модель объекта;

в) построить классы эквивалентности предметов;

г) отнести распознаваемый объект к какому-либо классу.

18. В результате обучения распознающая система должна приобрести способность:

а) реагировать различными реакциями на все объекты одного образа и различными - на все объекты различных образов;

б) реагировать различными реакциями на все объекты одного образа и одинаковыми - на все объекты различных образов;

в) реагировать одинаковыми реакциями на все объекты одного образа и различными - на все объекты различных образов;

г) реагировать одинаковыми реакциями на все объекты одного образа на все объекты различных образов.

19. Какие методы распознавания образов можно выделить:

а) метод перебора;

б) математический метод;

в) индуктивный метод;

г) нет верного ответа.

20. Система распознавания образов, предложенная Оливером Селфриджем называется:

а) пандемониум;

б) распознавание с помощью нейронных сетей;

в) геометрическая система;

г) экспертная система.

Список использованных источников

1 Братко, И. Алгоритмы искусственного интеллекта на языке Prolog / И. Братко.– 3-е издание.: Пер. с англ.– М.: Издательский дом “Вильямс”, 2004.– 640 с.

2 Джонс, М.Т. Программирование искусственного интеллекта в приложениях: практическое пособие / М.Т. Джонс; пер. А.И. Осипов. – 2-е изд. – Москва: ДМК Пресс, 2011. – 313 с.: табл., схем. –ISBN 978-5-94074-746-8; То же [Электронный ресурс]. –URL: http://biblioclub.ru/index.php?page=book&id=131005

3 Люгер, Д. Искусственный интеллект / Д. Люгер.– М.: Мир, 2003.–690 с.

4 Павлов, С.И. Системы искусственного интеллекта: учебное пособие / С.И. Павлов. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2011. – Ч. 1. – 175 с. –ISBN 978-5-4332-0013-5; То же [Электронный ресурс]. –URL: http://biblioclub.ru/index.php?page=book&id=208933

5 Павлов, С.И. Системы искусственного интеллекта: учебное пособие / С.И. Павлов. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2011. – Ч. 2. – 194 с. –ISBN 978-5-4332-0014-2; То же [Электронный ресурс]. –URL: http://biblioclub.ru/index.php?page=book&id=208939

6 Сидоркина, И. Г.Системы искусственного интеллекта: учеб.пособие для вузов / И.Г. Сидоркина. – М. :КноРус, 2011. – 245 с.

7 Смолин, Д.В. Введение в искусственный интеллект: конспект лекций / Д.В. Смолин. – М.: ФИЗМАТЛИТ, 2007. http://www.biblioclub.ru/book/76617/

8 Сотник С.Л. Проектирование систем искусственного интеллекта [Электронный ресурс] / С.Л. Сотник. – М.: Национальный Открытый Университет «Интуит», 2007.http://biblioclub.ru/index.php?page=book_view&book_id=234802

9 Спицын, В.Г. Представление знаний в информационных системах: учебное пособие / Спицын В.Г., Цой Ю.Р. – Томск: Изд-воТПУ, 2007. – 160 с.

10Чулюков, В.А.Системы искусственного интеллекта. Практический курс / Чулюков В.А., Астахова И.Ф., Потапов А.С. и др.под ред. И.Ф. Астаховой. – М.: БИНОМ. Лаборатория знаний, 2008.

11Ясницкий, Л.Н. Введение в искусственный интеллект: учеб. пособие для студ. высш. учеб.заведений /Л.Н. Ясницкий. – М.: Издательский центр «Академия», 2008. – 176 с.

12Яхъяева, Г. Э. Нечеткие множества и нейронные сети: Учебное пособие / Г. Э. Яхъяева. – М.: Интернет Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006. –316 с.

13www.intuit.ru,курс «Интеллектуальные робототехнические системы».

14www.intuit.ru, курс «Практическая информатика», Лекция 10 «Решение логических задач».

15https://vunivere.ru – учебные материалы для студентов, Новосибирский государственный технический университет, курс «Системы искусственного интеллекта».